# ROESKE'S COMPUTER MAG.

6.80 DM
60 öS
6.50 sFr
Januar 1988
1/88

Jetzt auch als Video
Information dazu im Heft

Featuring:
*Amiga & C-64*

**Das Magazin für Aufsteiger**

**Amiga**
★ Listings ★ Kurse ★ Reviews

**C64**
★ Kurs Programmiertechnik ★ Viele Listings

**Neu!**
★ Riesenspieleteil

**Neu!**
★ Video-Service

## C64 Spiele

| | CASS | DISK |
|---|---|---|
| AIRBORN RANGER | | 64,90 |
| CHAMONIX CHALLANGE | | 49,90 |
| CLEVER UND SMART | 34,90 | 44,90 |
| FOOTBALLER OF THE YEAR | 24,00 | 39,90 |
| FREDDY HARDEST | | 44,90 |
| GAME SET AND MATCH | | 59,90 |
| GARY LINEKERFOOTBALL | 34,90 | 49,90 |
| GUNBOAT | | 46,90 |
| HELLOWOON | | 44,90 |
| MEAN STREAK | | 44,90 |
| PIRATES | 39,00 | 52,00 |
| RAMPARTS | | 42,90 |
| RYGAR | 34,90 | 49,90 |
| SIDEWIZE | | 44,90 |
| STAR WARS | | 44,90 |
| STREET HASSLE | | 49,90 |
| SUPERSTAR ICE HOCKEY | | 49,90 |
| WINTER OLYMPIAD 88 | | 42,90 |

## Amiga Spiele

| | |
|---|---|
| AMIGA POWER PACK | 64,90 |
| BAD CAT | 59,90 |
| BALANCE OF POWER | 89,90 |
| BUREAUCRAZY | 84,90 |
| FEUD | 34,90 |
| FIRE POWER | 79,90 |
| GRID START | 34,90 |
| INDOOR SPORTS | 79,90 |
| INSANITY FLIGHT | 79,90 |
| KAMPFGRUPPE | 89,00 |
| KINGS QUEST TRIPLE PACK | 79,90 |
| LEVIATHAN | 64,90 |
| PINBALL WIZARD | 49,95 |
| Q-BALL | 64,90 |
| SWOPPER | 49,00 |
| TEST DRIVE | 98,90 |
| THAI BOXING | 34,90 |
| TRIVIA PROVE (DEUTSCH) | 34,90 |

# PREISHITS DES MONATS

| | | C64 | Amiga | | | C64 | Amiga |
|---|---|---|---|---|---|---|---|
| CALIFORNIA GAMES | 32,00 | 39,00 | 69,00 | WESTERN GAMES | 34,00 | 44,90 | 59,00 |
| ROADWAR EUROPA | | 59,00 | 59,00 | PHANTASIE II | | 59,00 | 59,00 |
| VERMEER | | 64,00 | | JAGD AUF ROTER OKTOBER | | | 74,90 |

## C64 Strategie

| | | |
|---|---|---|
| B-24 | | 64,90 |
| BISMARK | 32,90 | 46,00 |
| BLITZKRIEG | | 54,90 |
| DECISION IN THE DESERT | | 64,90 |
| GUADACANAL | | 49,90 |
| HALLS OF MONTEZUMA | | 79,90 |
| KAMPFGRUPPE | | 89,00 |
| OGRE | | 59,90 |
| OKINAWA | | 39,90 |
| PANZER GRENADIER | | 84,90 |
| POWER STRUGGLE | | 39,90 |
| ROADWAR EUROPA | | 64,90 |
| TOBRUK | | 49,90 |
| WAR GAME GREATS | 49,90 | 64,90 |
| WARSHIP | | 89,90 |

## Amiga Anwender

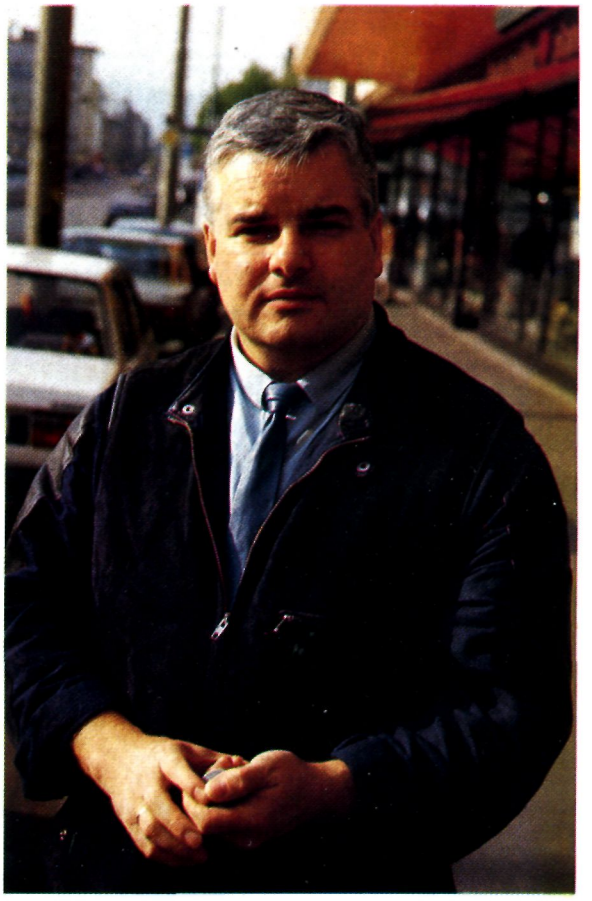
| | |
|---|---|
| AEGIS AUDIOMASTER | 129,90 |
| AEGIS DRAW PLUS | 498,00 |
| AEGIS SONIXVERSION 2.0 | 189,00 |
| BUTCHER 2.0 (DEUTSCH) | 139,00 |
| DELUXE MUSIC (DEUTSCH) | 298,00 |
| DELUXE PAINT (DEUTSCH) | 298,00 |
| DIGIPAINT | 169,00 |
| DRUCKERANPASSUNG CP-80X | 59,90 |
| DRUM STUDIO | 79,00 |
| GALILEO (ASTRONOMIE) | 179,00 |
| GO AMIGA DATEI | 199,00 |
| LOGISTIX (DEUTSCH) | 399,90 |
| SCULPT 3D PAL VERSION | 249,00 |
| SILVER | 349,90 |
| VIZAWRITE AMIGA | 198,00 |

** Wir suchen noch Programmautoren ** ** Preisänderungen vorbehalten **

VIZA SOFTWARE

CSJ COMPUTERSOFT GmbH * Händleranfragen erwünscht *
An der Tiefenriede 22 * 3000 Hannover 1 * Tel. Bestellservice (0511) 88 63 83
Riesenauswahl an Software *** sofort CSJ NEWS anfordern (Computertyp angeben) ***
Versand Inland: Vorkasse + 2,50 DM (Euroscheck in DM) per Nachnahme + 7,- DM

# Editorial



**Liebe Leserin, lieber Leser.**

Wäre es nicht schön, wenn wir über dieses Magazin das ganze Jahr 1988 in Verbindung bleiben könnten?
Eine ganze Menge Neuerungen hat dieses Jahr für Sie als Computer-User schon parat.
So haben Sie ab sofort Gelegenheit, einige Artikel aus diesem Heft, aber auch schon Tests und Reviews aus der nächsten Ausgabe in Form von bewegten Bildern mit Sprache und Sound auf Video zu erleben.
Wir wissen nicht, ob wir weltweit das erste monatlich regelmäßig erscheinende Homecomputer-Video anbieten - in Europa sind wir jedenfalls die ersten.
Den Sammlern unter Ihnen sei an dieser Stelle mitgeteilt: Wir haben nichts dagegen, wenn Sie sich eine Kopie für Ihr Archiv ziehen und diese auch Ihren Freunden vorführen. Solange Sie nicht damit Handel treiben ...
Bitte haben Sie Verständnis dafür, daß nicht jeder seine Kassette innerhalb einer Woche erhalten kann. Die in Umlauf gebrachte Menge ist so bemessen, daß alle Kunden innerhalb von drei Wochen beliefert werden können.
Wer es eilig hat, kann Roeske's Computer Video auch über seine Videothek ausleihen. Das ist bequemer und billiger. Es müssen aber mehrere Interessenten vorhanden sein, damit sich die Sache für den Videothekar lohnt. Wir haben die Videotheken informiert. Nun kommt es darauf an, ob genügend Leser diesen Weg wählen.
Mit dem Disketten-Service hat es einige Probleme gegeben. Aus unerklärlichen Gründen führte unser Kopierprogramm den Check, ob die Diskette fehlerfrei kopiert wurde, nach eigenem Gutdünken durch. Mal meldete es einen Error, dann war die Diskette fehlerhaft, mal meldete es keinen und die Diskette war trotzdem nicht in Ordnung. Obwohl dies nur 1 - 2 % aller Fälle ausmachte, ist es doch ärgerlich. Wir bitten dies zu entschuldigen und hoffen, daß wir in Zukunft von solchen Aussetzern verschont bleiben.
Eigentlich sollte in diesem Monat ja bereits die Topprogrammaktion starten, doch leider sind bisher noch keine wirklichen Topprogramme bei uns angekommen.
Naja, die Zeit zwischen dem ersten Aufruf und dem Druck dieses Heftes war wohl doch etwas zu kurz.

Übrigens; schauen Sie doch nochmal aufs Titelbild. Können Sie sich denken, warum wir gerade dieses Bild als Titelbild ausgewählt haben? Ich finde, es paßt zu unserem neuen großen Spieleteil. So ähnlich ging es nämlich in der Redaktion zu, als wir die die einzelnen Spiele benoteten.
Carsten Borgmeier, unseren Spieleressortleiter, kennen Sie vielleicht schon aus anderen Computerzeitschriften. Sein Ziel ist die absolute Aktualität, Ihnen die brandheißesten Neuerscheinungen zu präsentieren. Und das ist ihm auch gleich auf Anhieb gelungen. Einige Spiele sind so neu, daß Sie erst in diesen Tagen in die Läden kommen werden. Carsten hatte sie sich direkt bei den Programmierern und Herstellern besorgt und die Tests noch zwei Tage vor Druckbeginn durchgeführt.
Ich wünsche Ihnen viel Glück im Neuen Jahr und viel Spaß beim Lesen

Ihr Ralph Roeske
(Chefredakteur)

# ROESKE'S COMPUTER VIDEO

monatlich

1/88

**Featuring:**
*Amiga & C-64*

## Das erste Homecomputer-Video-Mag. Europas

**System:** VHS
**Mietpreis:** 12,- + 3,- Porto
**Mietdauer:** 48 Std.
**Magazinlänge:** ca. 90 Min.
**Im Versand durch uns!**

**oder:**

**Mietpreis:** 8,-DM - 12,-DM
**Mietdauer** 24 Std.
**In Ihrer Videothek!**

Sollte Ihre Videothek (was bei Erscheinen des Videos noch vielerorts der Fall sein wird) Roeske's Computer Video noch nicht vorrätig haben, so zeigen Sie doch bitte diese Zeitschrift und unsere Telefonnummer. Der Videothekar wird sich dann an uns wenden.

**Durch das Programm führen Chefredakteur Ralph Roeske und Susan Bartholomäus (Bild Mitte).**

# Was bringt Roeske's Computer Video?

Kurz gesagt: Alles, was Sie in Roeske's Computer Mag. finden! Nur ist vieles über das Medium Fernsehen bzw. Video viel ausführlicher zu vermitteln. Stichwort Buch-Reviews. Was in RCM nicht möglich ist, in RCV geht's: Ein Durchblättern mehrerer Bücher mit Stop auf wichtigen Seiten, damit Sie richtig im Bilde sind. Stichwort Software-Reviews. In RCM sehen Sie von jedem Spiel einen Schnappschuß eines Levels - in RCV sehen Sie, wie die Spiele ablaufen - ausführlich in vielen Levels, so weit wie Garry, Carsten und unsere anderen Tester es schaffen. Sie sehen wie man mit einem DeskTopPublishing-Programm arbeitet, Seiten erstellt und druckt. Sie sehen wie Digiview ein Bild scannt. Sie hören(!!!) wie bei Testdrive die Motoren dröhnen, wie bei Bad Cat die Biergläser über den Tresen rutschen, wie bei Fire Power die Hubschrauber dem Panzer auf die Pelle rücken, Sie hören, wie PerfectSound Töne digitalisiert und wie sie wiedergegeben werden!

Außerdem stellen wir vor: Die Programme aus dem Heft; Public Domain Software; ABC-Kurse (Assembler, Basic, C). Das monatlich erscheinende Video-Magazin enthält in lockerer Folge außerdem Interviews (z.B. bei Rainbow Arts, Activision, Commodore, usw.). Wir bringen Messe-Rundgänge, die wir im In- und Ausland für Sie aufnehmen u.v.m!

Nutzen Sie dieses neue Medium.

Bestellen Sie sich Ihre Copy mit der Zahlkarte von Seite 78

# Inhalt

## Spiele-Reviews


SQIJ 7
Test Drive 9
To Be On Top 12
Combat School 13
Crazy Cars 14
Buggy Boy 14
Mean Streak 15
Ramparts 15
Skate or die 16
Guadacanal 17
Clever & Smart 17
Super Star Soccer 18
Winter Olympiad 88 19
Action Force 19
The Great Giana Sisters 20
Ninja Mission 21
Space Ranger 21
Feud 22
Galactic Games 22
Star Wars 23
Ralley Master 23
S.D.I. 24
Grid Start 24


## Kurse


Umsteigen leicht gemacht (Teil 3) 26
C für Einsteiger (Teil 2) 28
Cursus infinitus (C64) 66


## Programmiertechnik


DiskInfo 32
Display Alert 33
ChangeName 33
DigiSound 34
OwnScreen 34
WorkbenchSize 36
Neue KS-Routinen 43
Fast Line 69


## Listings


Critical Mass (Amiga) 30
Bobeditor (Amiga) 39
Nimbi (Amiga) 46
Börsenspiel (Amiga) 52
17 und 4 (Amiga)53
Unheimliche Begegnung (C64) 56
Disks of Tron 56


## Rubriken


Editoral 3
Brandheiße Spiele-News 6
Top Ten »Amiga & C64« 11
Gewinnspiel 50
Programmierwettbewerb 77
Impressum 78
Disketten- & Video-Service 79

# Brandheiße Spiele-News

**Hallo, lieber Spielefreund !**

**TATARATA! Roeske's Computer Mag. führt ab dieser Ausgabe einen aktuellen Spieleteil für C64 und Amiga ein.**
**Mit topaktuellen Reviews, heißen Informationen, Wettbewerben, Interviews mit Programmierern, Spieletips (z.B. Adventureauflösungen, Pokes, Karten und Kurzhilfen), Nachrichten aus der Spielszene und Berichten von internationalen Homecomputerspiele-Messen versuchen wir Sie jeden Monat auf 15-20 Seiten zu informieren und zu unterhalten.**
**Als langjähriger Spieletester und leidenschaftlicher Spielefreak habe ich diese Aufgabe übernommen.**
**Hunderte von Spielen durfte ich bisher testen und ich kann Ihnen versichern, ich bin ihrer nicht überdrüssig geworden. Im Gegenteil! Viele Spiele ziehen mich in ihren Bann, so daß ich oft alles um mich herum vergesse. Aber keine Angst, auf Ihren Spieleteil brauchen Sie nun nicht mehr verzichten.**
**Übrigens: Ich habe keine prinzipielle Abneigung gegen eine bestimmte Spielegattung. Ich mag Ballerspiele a la 'Rambo'genauso gerne wie Sportspiele oder Adventures.**
**Aus den vielen Spielen, die hier monatlich bei mir eintrudeln, kristallisieren sich meist ein oder zwei Programme heraus, die besonders gut oder besonders schlecht sind. Wir haben für derartige Spiele extra zwei Rubriken eingeführt. Unter der Rubrik RCM Renner wird das beste Spiel des Monats beschrieben, während wir unter der Rubrik RCM Penner den softwaremäßigen Müll testen, den es im jeweiligen Monat auf dem Spielemarkt gibt.**
**Unsere Bewertungsskala reicht von 0 - 7. 0 ist natürlich die schlechteste Wertung, 7 die beste.**

**Also, viel Spaß beim Lesen, wünscht Ihnen**

**Carsten Borgmeier**
**(Spieleressortleiter)**

## Compilations

Zur Weihnachtszeit sind Spielesampler oder Compilations, wie sie auch genannt werden, sehr beliebt. Die Samplers kosten im Durchschnitt DM 40,- bis DM 50,-; dafür hat der Computerspielefreak aber manchmal bis zu 8 Computerspiele zum Preis von einem.

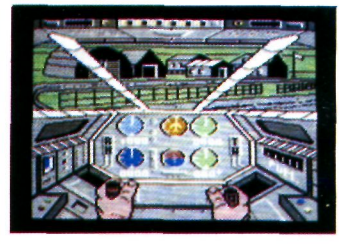
**Ocean** präsentiert eine neue Compilation mit Namen **Game Set and Match**. Wie der Name schon eingibt, sind auf dieser Spielesammlung ausschließlich Sportspiele zu finden. Darunter befindet sich das erstklassige Tennisspiel 'Matchpoint', World Series Baseball, Jonah Barrington Squash, was übrigens eins der ersten Spiele mit deutlicher Sprachausgabe war. Außerdem gibt es noch ein Billiardspiel namens Snooker, den Tischtennisspaß Ping Pong, die beiden Olympiasportarten Hyper Sports und Daley Thompson's Supertest. Ferner das Fußballspiel Super Soccer, Two on Two Basketball und last but not least Championship Boxing. Für Sportspielfans ist Game Set Match eine lohnende Anschaffung.
**Bezugsquelle:**
**CSJ, Hannover**

Game Set and Match ist aber nicht die einzige Compilation, die **Ocean** zu bieten hat, Es gibt auch noch **Magnificent Seven**, auf der so einzigartige Spiele wie **Head over Heals**, ein 3D-Kletterspiel mit vielen verschiedenen Räumen. Natürlich ist auch was für harte Männer dabei: **Stallone Cobra**. Sie laufen in diesem Spiel von links nach rechts und ballern mit Ihrer Kanone auf alles, was sich bewegt. **Short Circuit** heißt ein gut gemachtes Spiel mit einem Roboter. Das Game lehnt an die Handlung des Films 'Nummer 5 lebt' an. **Frankie goes to Hollywood** ist schon ein Oldie unter den Computerspielen. Trotzdem spielt es sich noch sehr gut. Mit **Arkanoid** fing der Breakoutboom an. In diesem Klassespiel von **Imagine** müssen Sie mit einem Ball eine Mauer, die aus vielen Steinen besteht, abräumen. Ein Wahnsinns-Spaß. Bei **Wizball** sind sich alle Spieletester einig: Ein Klassespiel. Sie haben bestimmt auch viel Spaß daran, einen Kürbis durch eine wundersame Welt zu steuern. Spiel Nummer 7 ist indiziert, deshalb: Kein Kommentar. **Yie ar Kung Fu** wird noch als Bonusspiel hinzugegeben. Magnificent 7 ist eine Compilation, auf der es kein schlechtes Computerspiel gibt. Die Spiele sind keine Ladenhüter, sondern Hits des letzten Jahres. Also zugreifen, solange der Vorrat reicht.
**Bezugsquelle:**
**Seemüller, München**

**Ocean** bietet auf einer weiteren Compilation fünf Actionskriegsspiele, von denen bis auf **Top Gun** und **Army Moves** alle indiziert sind.

**Elite Systems** hat auch eine interessante Compilation zu bieten. **6-Pak Vol. 2**heißt die überaus interessante Spielesammlung, auf der bekannte Titel wie Flugzeugsimulator **Ace**, Karatespiel **International Karate**, Actionskriegsspiel **Into the Eagles Nest** Weltraumballerspiel **Lightforce** und **Shockway Rider** vertreten sind.
**Bezugsquelle:**
**Elektronik von A-Z**

**Imagine** wirft im Weihnachtsrummel auch eine Compilation mit Namen **Imagine's Arcade Gigants** auf den Markt. Auf diesem Sampler sind die Spiele **Arkanoid**, **Galvan**, **Game over**, **Mag Max**, **Slap Fight** und **Terra Cresta** zu finden.



**U.S. Gold** bringt eine hervorragende Compilation heraus. **Solid Gold** enthält Namen, die das Herz eines jeden Spielefans höher schlagen lassen. Da gibt es die Sportspiel Olympiade **Winter**

**Games**, das Labyrinthspiel aus der Spielhalle **Gauntlet** (Urvater von Garrison), das exellente Golfspiel **Leaderboard** und das spannende Agentenepos **Infiltrator**, bei dem Actionadventure und Flugsimulator hervorragend gekoppelt wurden.
**Bezugsquelle:**
**Profisoft**

Von der Spielesammlung **Power Pack** für den Amiga war ich etwas enttäuscht. Es wurden wirklich nur Billigproduktionen auf die 3 Amigadisketten verbannt. Die beiden Brettspiele **Backgammon** und **Memory**, ein Bilderpuzzle, ein weiteres Brettspiel mit dem Namen **One Check** und zwei Ballerspiele machen schneller müde, als eine Schlaftablette.
**Bezugsquelle:**
**Foto Quelle, Delmenhorst**
**Profisoft**.

Durch den Erwerb einer Spielecompilation können Sie sogar ein gutes Werk tun. Die Benefizkassette des **Englischen Kinderschutzbundes** macht es möglich. Die C64 Variante bietet Spiele wie: **Barry Mc Guigan's World Championship Boxing, Bounces, Bounty Bob strikes back, Elektraglide, Monty on the Run, Starion, Xeno, Prodigy** und ein indiziertes Spiel versprechen eine Menge Spielspaß.
**Bezugsquelle:**
**Seemüller, München**

Nicht zu verwechseln mit Power Pack ist **Power Plays**. Eine Compilation von Low Budgetspielproduzent **Power House**. Spiele wie **Herkules, Terminator, Aftermath, Space Warrior, Gods an Heros, Xenon Ranger, Gun Runner** und **SQIJ** sollen für ca. 20,- DM den Besitzer wechseln. Die Kassette sollte man lieber in den Mülleimer werfen und nicht in eine Kassettenhülle stecken. Von dem originellsten Spiel **SQIJ** bieten wir Ihnen hier einen Kurztest:

**Alpha Omega Software** beschert uns für ungefähr DM 10,-- ein recht hübsches Programm mit der dämlichsten Anleitung des Monats.
Ihre Aufgabe ist es, SQIJ, was nach Aussagen der Anleitungen 'Squidge' heißt (sind Sie jetzt schlauer?) durch ein unerforschtes großes unterirdisches Höhlensystem zu steuern.
Irgendwo in diesem riesigen unterirdischen Höhlensystem befindet sich eine verlassene Höhle mit Namen 'Luut-too-Weith' 'oder so ähnlich), die Sie unbedingt finden müssen.
Ja, denn da können Sie nämlich den Enertrie bauen, der dem Helden des Spiels Unsterblichkeit verleiht. Dieser Enertrie (was immer das auch sein soll), besteht aus 6 verschiedenen Teilen, die irgendwo im Höhlensystem versteckt sind. Sie können mal raten, wessen Aufgabe es ist, die Teile zu finden.
Richtig, das ist ihre zweite Aufgabe. Damit aber nicht genug. Das ganze Höhlensystem ist voller Monster, die Ihnen die Unsterblichkeit nicht gönnen wollen.
Zum Glück haben Sie aber Ihren 'Splurger' bei sich. Mit ihm können Sie die Monster eleminieren.
Der Held des Spiels, also SQIJ, hat starke Ähnlichkeit mit einem Vogel. SQIJ ist aber kein Vogel, sondern ein Squidge.
Was auch immer, Ihr Squidge hat eine Energieanzeige. Sie sollten daher Zusammenstöße mit Monstern und Felswänden vermeiden.
Doch das ist leichter gesagt, als getan. Von allen Seiten kommen Monster auf Sie zugeflogen. Sie können gar nicht immer alle abschießen.
Ich halte die gestellte Aufgabe für unmöglich zu erfüllen. Bei mir setzte sich nach einer Stunde der Frust durch. Ab und zu traf ich während meines Tests auf Energietüren, die man nur öffnen kann, wenn man einige im Labyrinth herumliegende Früchte einsammelt, diese dann zu Disperna-Monstern bringt, die einem dann einen Passierschein aushändigen, mit dem man ungehindert durch die Energietüren gelangen kann.
Mir war es leider nicht möglich, durch die Energietüren zu gelangen, da ich nicht weiß, wie Disperna Monster aussehen. (Wissen Sie es?).
Die Grafik des Titelhelden SQIJ ist recht gut gelungen, ebenso die Grafik der Monster; die einzelnen Höhlen sind bloß sehr lieblos gestaltet. Der Sound ist nicht schlecht gemacht, reißt aber niemanden vom Hocker. Spaß hatte ich nur beim Lesen der dämlichen Anleitung. Zur Motivation läßt sich soviel sagen: Jeder wird schon sehr bald erkennen, wo seine Grenzen sind.

**SQIJ**
**System: C64**
**Hersteller: Power House**

| | |
|---|---|
| **Spaß/Motivation** | ★ 1 |
| **Grafik** | ★★★★ 4 |
| **Sound** | ★ 1 |
| **Originalität/Idee** | ★★★ 3 |
| **Schwierigkeit** | ★★ 2 |
| **RCM-Wertung** | ★★ 2 |

## Low Budgetspiele

**Alternative**, im deutschen Vertrieb von **Profisoft**, veröffentlicht das Adventure **Football Frenzy** für Fußballfans.
Gegenüber allen Erwartungen haben Sie es geschafft, das Cup Finale zu erreichen. Es sind nur noch wenige Tage bis zum Endspiel. Im Moment ist noch alles in Ordnung, aber wird das so bleiben?
Viele Schwierigkeiten erwarten den Spieler von 'Football Frenzy'. Das Adventure spielt sich sehr schwer, da der Parser nur ca. 160 Wörter versteht.
**Bezugsquelle:**
**Elektronik von A-Z**

'Time Trax' hieß ein Vollpreisspiel, in dem man einen Zeitreisenden zu steuern hat. **Bug Byte** veröffentlicht **Time Trax** als Low Budgetspiel.

Zum Preis von DM 14,95 bietet **Mastertronic** die Spiele **Saracen, Tennis** und das Autorennen **Talledega**, das vor Jahren schon einmal unter dem Namen 'Richard Petty's Rennzirkus' auf dem Markt erschien.
Dramatische Überholmanöver auf der Piste gehören genauso zu Talledega, wie Boxenstops.

**Alternative Software** hat noch zwei Low Budgetspiele auf der Pfanne. Das eine nennt sich **Danger Mouse in Double Trouble**. Sie steuern eine Maus durch einen Dschungel und müssen eine Menge Abenteuer bestehen.
Das zweite Low Budgetspiel von Alternative hat den Namen **Dead or Alive**. In diesem Spiel müssen Sie als Sheriff Gangster abknallen.

## Programmieren Sie Selbst?

**Dann senden Sie uns doch mal etwas zu!**
Informationen dazu finden Sie in dieser Zeitschrift.

# Brandheiße Spiele-News

**KGB Agent** ist der Titel eines grafisch recht gut gestalteten Actionspieles, in dem Sie als KGB Agent amerikanische Soldaten niedermetzeln müssen.

**Grand Prix Simulator** von **Codemasters**, ist die billigste Variante der Supersprint- oder Ralley-Master-Idee. Zwei oder (im Zweierspielmodus sogar drei) Autos rasen in einem Rondell um die Wette. Digitalisierte Sprachausgabe!

**Venom**, ein Zwei-Wortparser-Adventure, **Angleball**, eine Art Billiardspiel und das Ballerspiel **Starforce Fighter**, sind ebenfalls neue Low Budgetspiele von **Mastertronic**.

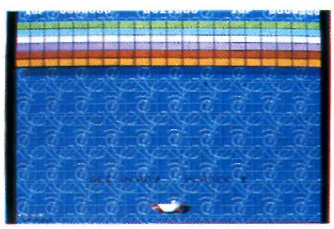
In **Mission Genocide** von **Firebird 199 Silver Range**, müssen Sie einen Planeten vernichten, bevor er Sie vernichtet. Na, dann mal ran an den Feind!

**Haben Sie Fragen zu den Spielen, zum Amiga, zum C64. Wir beantworten Sie Ihnen gerne. Schreiben Sie uns ein Kärtchen und wir bemühen uns, die Antwort einzuholen, sofern wir sie nicht selbst beantworten können. In einer kleinen Rubrik zu diesem Thema werden wir diese Fragen beantworten.**

'Garrison' von **Rainbow Arts** ist so ein großer Erfolg geworden ist, daß die Programmierer wieder an die Arbeit gegangen sind, den Fortsetzungstitel **Garrison II** zu programmieren. 140 neue Level und neue Feinde werden dem Spieler im neuen Garrison überraschen. (Testbericht folgt in der nächsten Ausgabe cb). Garrison II gibt es bald für den Commodore Amiga.

Zehn verschiedene Landschaften haben Sie bei **Super-G-Man** von **Code Masters** als Astronaut mit Raketenantriebsrucksack zu durchreisen, bevor Sie mit einem Raumschiff zur Erde zurückfahren können.

**Micro Partner** präsentiert eine tolle Olympiade aus dem wilden Westen - **Western Games**. Amiga und C64 Freaks werden viel Spaß mit den lustigen Disziplinen haben: Armdrücken, Bierschießen, Kautabakspucken, Tanzen, Kuhmelken und Wettessen sind die Salondisziplinen. Die Amigaversion bietet viele digitalisierte Sounds. Beim Wettessen ist beispielsweise ein waschechter Rülpser zu hören.



**Amegas** nennt sich die 8945. Breakoutversion von **Reline**. Ziel des Spiels ist es immer noch, mit Hilfe eines Balls und eines Schlägers, Steine aus einer Wand zu brechen.
Gesteuert wird mit der Maus. Natürlich ist Amegas für den Amiga. Der Sound ist übrigens vom besten, was bisher auf diesem Computer komponiert wurde.
**Bezugsquelle:**
**Soyka Datentechnik, Bochum**
**Profisoft**

**Hotline**
**☎ 05651-8778**

**Jeden Donnerstag sind wir von 17.00 bis 19.00 Uhr für Sie am Telefon, um Ihre Fragen zu beantworten.**

# Brandheiße Spiele-News

**Test Drive** heißt eine sehr realistische Fahrsimulation der Firma **Accolade**. Sie wollen ein Auto kaufen und können mit den Autos aus dem Geschäft ein paar Proberunden drehen. Bei Ihren Testfahrten ist auf Geschwindigkeitsbegrenzungen und auf die Polizei zu achten.
Bei den Fahrzeugen handelt es sich um einen Porsche 911 Turbo, Ferrari Testarossa, Lotus Esprit, Lamboghini Countach und einen Chevrolette Corvette Sting Ray.
Interieur und Cockpit der Fahrzeuge sind exakt nachgebildet. Die Detailfreudigkeit der Grafik-Designer dieses Spiels geht soweit, daß selbst die Schaltung, die bei jedem Schaltvorgang eingeblendet wird, dem Original entspricht.
Im Rückspiegel sieht man nicht

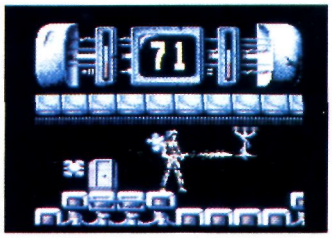
nur Fahrzeuge, die sich von hinten nähern, sondern auch die Autos, die einem entgegenkommend passiert haben.
Als alter Autonarr bin ich selbstredend von dieser Simulation begeistert. Hoffentlich kann der 'Händler' bald weitere Fahrzeuge anbieten. Ich schaue dann gerne mal rein, um eine Fahrt mit einem Jehle Pantera GT5, oder einem Koenig Specials Mercedes zu unternehmen. RR

**Bezugsquelle:**
**DTM , Wiesbaden**
**Soyka Datentechnik, Bochum**
**LaSch, Oberhausen**

| Test Drive | |
|---|---|
| System: Amiga, C64 | |
| Hersteller: Accolade | |
| Spaß/Motivation | ★★★★★★ 6 |
| Grafik | ★★★★★★★ 7 |
| Sound | ★★★★★ 5 |
| Originalität/Idee | ★★★★★★★ 7 |
| Schwierigkeit | ★★★★★★ 6 |
| RCM-Wertung | ★★★★★★ 6 |

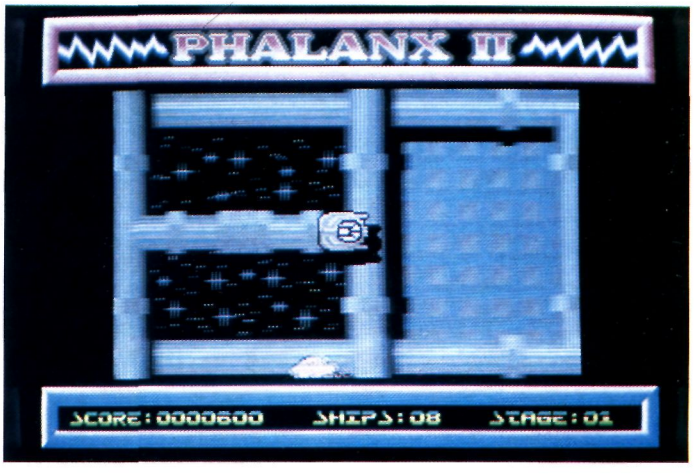
**Activision** hat die Rechte zur Produktion des Computerspiels **Predator**, basierend auf dem gleichnamigen Film erworben.
Mit Arnold Schwarzenegger in der Hauptrolle spielte Predator in den ersten drei Tagen 12 Millionen Dollar ein.
Major Dutch Schäefer ist Berufssoldat. In seiner Welt sind die Grenzen zwischen Freund und Feind klar gezogen - bis er seine Männer in eine Mission führt, in der nur das Grauen regiert und seine Regeln keine Geltung mehr finden.
Er und seine Elitetruppe haben den Auftrag eine Gruppe von Verbündeten aus den Fängen der Guerrillas im lateinamerikanischen Urwalds zu befreien. So sein Auftrag im Film.
Gegenwärtig wird Predator, die Computerversion, in Zusammenarbeit mit System 3 entwickelt.
Der Spieler übernimmt die Rolle von Dutch Schaefer und muß den Predator vernichten.
Doch bevor er mit dem Raubtier in Menschengestalt, einem Feind, der sich dem Dschungel anpaßt, und mit unmenschlicher Kraft zuschlägt, konfrontiert wird, muß er verschiedene Levels des Dschungels mit allen seinen Feinden überwinden.
**Systeme:**
**Amiga, C-64**

**U.S Gold**'s neues Lable **Go** hat ein hervorragendes Actionspiel mit Namen **Trantor** geschaffen. Grafik und Sound sind echt phänomenal. (C64)
**Bezugsquelle:**
**Profisoft**




**Seven Minutes To Midnight**, sieben Minuten bis Mitternacht, so heißt ein "Spiel" gegen die Zeit, in der es gilt, einen Atomkrieg zu verhindern. Erhältlich für Commodore C64 und Amiga

Ein neues Ballerspiel aus dem Hause **Kingsoft** ist **Phalanx II**. Sie fliegen ein Raumschiff von der linken zur rechten Bildschirmhälfte und schießen dabei feindliche Raumschiffe ab.
Nach jedem Level gibt es eine Bonusrunde, in der man sich einem Bonusspielchen widmen kann.




Sie werden mir nicht glauben, was für ein Spiel das ist. Richtig, eine Breakoutvariante.
Wenn Sie eine Kugel verpaßt haben, können Sie im ganz normalen Spiel wieder auf feindliche Raumschiffe ballern. Öfter mal was Neues. (Amiga)

# Brandheiße Spiele-News

Fernöstliche Kampfsportspiele scheinen hoch im Kurs zu sein. **Little Dragon** heißt ein neuer Vertreter dieses Genres.
Die Grafik und der Sound sind zwar in Ordnung, der Spielwitz kommt bei 'Little Dragon' aber leider zu kurz. (Amiga)

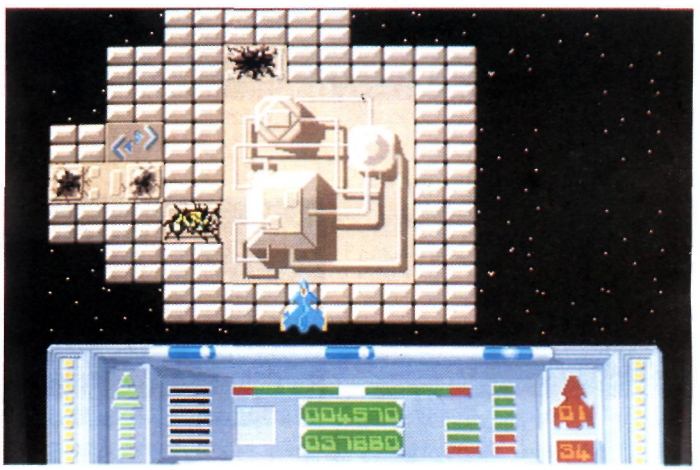
**Hunt for the Red October** ist eine neue U-Boot-Simulation von **Argus Press Software**. Als Kommandant eines russischen U-Boots haben Sie beschlossen, in den Westen überzulaufen.
Praktisch als Mitbringsel wollen Sie dem Westen ihr U-Boot anbieten. Das mögen die Russen aber gar nicht gerne. Man macht Jagd auf Sie. Mast- und Schotbruch! (Amiga)
**Bezugsquelle:**
**Elektronik von A-Z, Berlin**

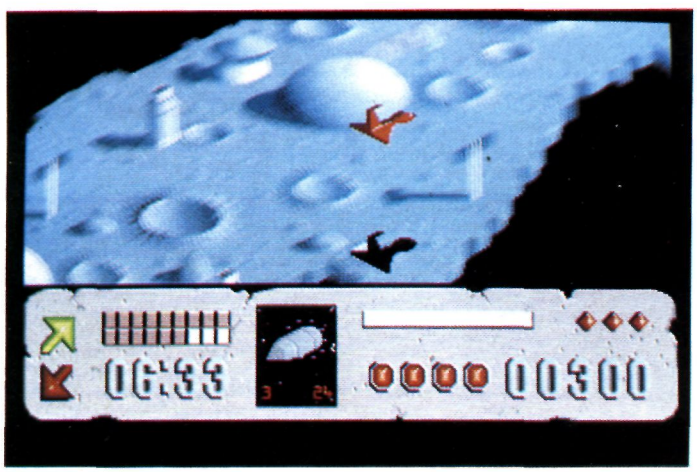
**Darkcastle** nennt sich ein tolles Kletterspiel. Es gibt eine Menge Level zu entdecken. (Amiga)

**Rygar** heißt ein neues Spiel von **U.S. Gold**. Ein totaler Flop! Miese Grafiken und eine Handlung für Joystickproleten. Ballern, ballern und nochmals ballern, auf alles, was sich bewegt.

Ein neues Spiel aus der Sparte fernöstliche Kampfsportarten hat Anco zu bieten. Zwei Kämpfer müssen sich gegenseitig solange auf dem Ommen hauen, bis einer tot umfällt. Mittel zum Zweck sind lebensgefährliche Karatetritte. (Amiga)

Insanity Fight' ist ein neues Spiel für den Amiga. Sie steuern ein Raumschiff und müssen Aliens abschießen, die sich in großen Scharen auf Sie stürzen. (Amiga)



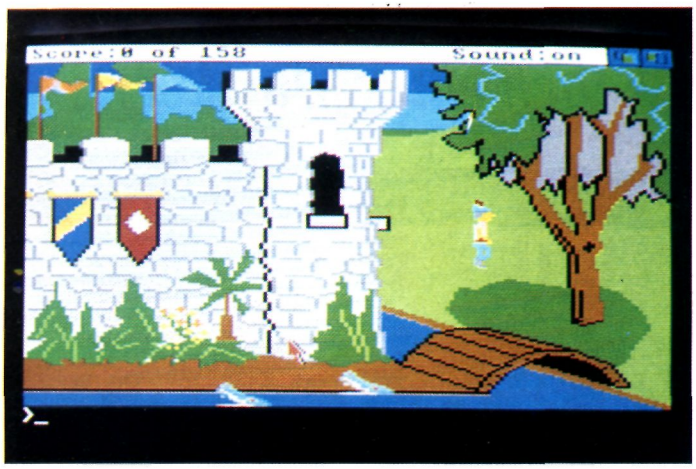


Ein neues Ballerspiel von English Software: 'Leviathan'!
Sie steuern einen Abfangjäger und fliegen über eine dreidimensionale Landschaft. Dabei dürfen Sie nach Herzenslust ballern. (Neu auf Amiga)

**Wollen Sie mehr über die in diesem Heft vorgestellten Titel wissen? Dann bestellen Sie sich doch Roeske's Computer Video. Aus der Masse der Spiele haben wir die besten herausgesucht, um Sie Ihnen auf Videoband mit Sound und Action in vielen Leveln zu präsentieren.**

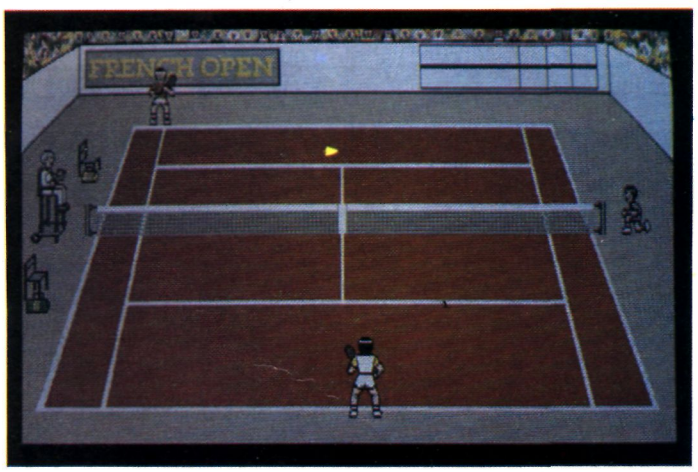
**Mehr dazu auf den Seiten 4 und 79.**

# Brandheiße Spiele-News




**King's Quest** heißt ein neues Adventurepaket von **Sierra Online**. Sie können den Held des Spiels mit dem Joystick durch die sehr große Adventurewelt steuern. Wenn er aber beispielsweise eine Tür öffnet, muß diese Aktion über Keyboard eingegeben werden.

Kings Quest - eine neue Perspektive für Adventures. Der Parser weiß auf jede Eingabe eine logische Antwort. (Amiga)




Wollten Sie nicht immer schon einmal Boris Becker nacheifern? Ja? Dann ist 'Grand Slam Tennis' genau das Richtige für Sie. Gespielt wird über Maus. Eine gute Grafik läßt Tennisatmosphäre aufkommen.

## Activision PCW Show

Am 17. November lud Activision zur Activision PCW Show ins Colonius Turmrestaurant zu Köln ein.

Pressevertreter, Publisher, Distributoren, Händler, User und Programmierer waren anwesend.

Auf der Party wurden Activisions neue Produkte vorgestellt. Sehr interessant erschien mir das Adventure **Maniac Mansion** in deutscher Sprache. Bei diesem tollen Adventure hat der Spieler die Aufgabe, seine Freundin aus den Klauen eines verrückten Wissenschaftlers zu befreien. Verrückte Cartoonszenen, urkomische Texte und eine intelligente Menütechnik machen Maniac Mansion zu einem Erlebnis.

Desweiteren wurde **Rampage** vorgestellt. Ein Spiel, bei dem drei Spieler gleichzeitig ihre destruktiven Energien freisetzen können. Sie dürfen auf Gebäuden herumtrampeln und sie dem Erdboden gleichmachen, denn dafür gibt es Punkte.

Man konnte sogar schon an einem der zahlreichen Vorführcomputer eine Vorabversion von **Bangkok Knights** spielen. Es war zwar nur der erste Level fertig, aber der zeigte schon das Können des Programmiererteams.

Alle Spiele wurden über eine große Kinoleinwand präsentiert. Frau Ewaldsen, (Pressesprecherin bei Activision), führte durchs Programm.

Sie zeigte Herrn Roeske und mir am späten Abend noch ein neues Spiel auf dem IBM (siehe Foto). Der Abend war gelungen (die letzten Gäste gingen erst lange nach Mitternacht), auch wenn bei der Programmpräsentation eine Vorführversion abstürzte.

**Schnappschuß von der Activision-Show (von links: Ralph Roeske, Carsten Borgmeier und Katrin Evaldsen)**

# Die RCM-Computer-Games-Parade

An dieser Stelle möchten wir Ihnen immer eine aktuelle Verkaufshitparade von deutschen Großhändlern präsentieren.

Da uns bei dieser Ausgabe leider zu wenig Zeit blieb, noch eine solche in Erfahrung zu bringen, haben wir eine Liste der persönlichen Lieblingstitel einiger Schul- und Berufsschulklassen für C64 und Amiga zusammengestellt:

## TOP TEN C64

| | | |
|---|---|---|
| 1. | Gary Lineker's Superstar Soccer | (Gremlin Graphics) |
| 2. | Great Giana Sisters | (Time Warp Productions) |
| 3. | Rampage | (Activision) |
| 4. | Skate or die | (Electronic Arts) |
| 5. | Airborne Ranger | (MicroProse) |
| 6. | Street Sports Basketball | (Epyx) |
| 7. | International Karate + | (Activision) |
| 8. | To be on Top | (Rainbow Arts) |
| 9. | In 80 Days Around The World | (Rainbow Arts) |
| 10. | Bubble Bobble | (Firebird) |

## TOP TEN AMIGA

| | | |
|---|---|---|
| 1. | Gridstart | (Anco) |
| 2. | King's Quest | (Sierra) |
| 3. | Test Drive | (Accolade) |
| 4. | Space Ranger | (Mastertronic) |
| 5. | Emerald Mine | (Kingsoft) |
| 6. | Ninja Mission | (Mastertronic) |
| 7. | Western Games | (Micro Partner) |
| 8. | Instant Fight | (Microdeal) |
| 9. | Crazy Cars | Titus |
| 10. | Goldrunner | (Microdeal) |

# Brandheiße Spiele-Reviews

## Der Renner des Monats

# To Be On Top: Aus dem Leben eines Musikers

**Erstklassige Spiele müssen nicht nur aus England oder den USA kommen. Das man es auch in Deutschland versteht, Spielsoftware zu entwickeln, die auf dem internationalen Markt mithalten kann, beweist die Gütersloher Softwarefirma Rainbow Arts.**

Geschäftsführer Marc Ullrich und seine 60 Mitarbeiter freuen sich über den Erfolg, den sie Spielen wie **Garrison** (zur Zeit meistverkauftes Amigaspiel in Deutschland), **Bad Cat, Jinks, Street Gang, Baby of Can Guru** und **In 80 Days around the World** zu verdanken haben.

Ein Name taucht im Zusammenhang mit Rainbow Arts immer wieder auf: Chris Hülsbeck. Chris, seines Zeichens hervorragender Soundprogrammierer, komponiert spektakuläre Sounds, die das letzte aus dem C 64 herausholen.

Fünfstimmige Musikstücke, Gesang oder digitalisierte Drums sind für ihn ein Kinderspiel. Nachdem er für die letzten Rainbow Arts-Produkte 'nur' den Sound programmiert hatte, wagte er sich endlich an sein eigenes Spiel.

'To be on Top' heißt das Meisterwerk, das mit hundertprozentiger Sicherheit ein Hitparadenstürmer wird.

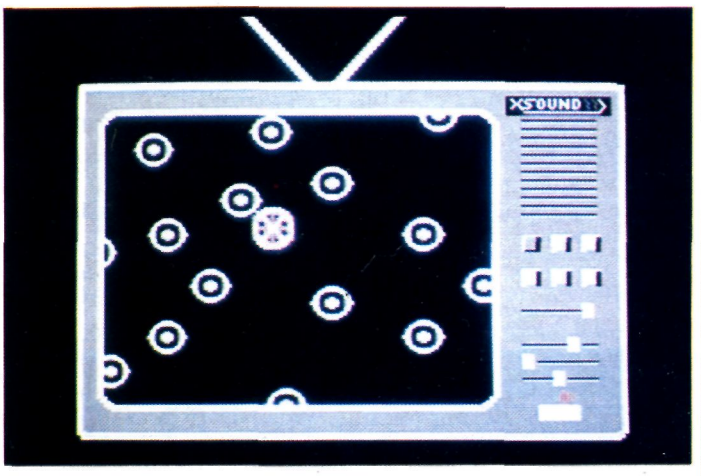
Und um die Erstürmung einer Hitparade geht es auch in diesem neuartigen Spiel:

Sie schlüpfen in die Rolle eines armen Schluckers, der nichts anderes gelernt hat, als Musik zu machen. Einen Hit müßte man landen, dann wären alle Probleme beseitigt. Nur leider will unserem Musikus nichts Rechtes einfallen.

Eines Abends, der Held kommt gerade von einem Besäufnis aus der Kneipe um die Ecke, da fällt der Groschen.

Eilig steuern Sie den Möchtegernstar über die Straße. Aufgepaßt: Eine wütende Rentnerin droht mit Ihrem Krückstock, Sie zu verprügeln. Ein Musikfreak mit einem Ghettoblaster sucht ebenfalls Streit. Sie sollten auch auf Panzer und einen komprimierten Blechkasten mit viel Rädern achten.

Schnelle Joystickreaktionen sind gefragt. Wenn es Ihnen gelungen sein sollte, den Musikus wohlbehalten nach Hause zu manövrieren, folgt das nächste Problem. Mutti sitzt auf dem Fußboden und schrubbt.




Der angehende Superstar achtet darauf, seiner Mutter nicht zu nahe zu kommen. Diese ist nämlich auf ihren Sohnemann sauer, weil er so spät nach Hause kommt.

Gelingt es, an der Mutter vorbeizukommen, ist Fernsehzeit. Sie setzen sich vor die Glotze, denn das inspiriert. Mit einem Fadenkreuz müssen Sie versuchen, verschiedenfarbige Inspirationspunkte zu sammeln, die allesammt über den Bildschirm huschen.

Bewegt man das Fadenkreuz über die Inspirationspunkte und drückt den Feuerknopf, wird man inspiriert. Da der zu erstellende Superhit vierstimmig sein soll, gilt es, vier Inspirationen pro Stimme einzusammeln.

Hat man die nötigen Inspirationen gesammelt, setzt man sich ans Klavier, um zu komponieren. Auf dem Bildschirm sind eine Klaviertastatur und die Inspirationspunkte zu sehen. Aus jeder Stimme muß man mit Hilfe einer joystickgesteuerten Hand eine Inspiration heraussuchen, von der man glaubt, daß sie am besten klingt.

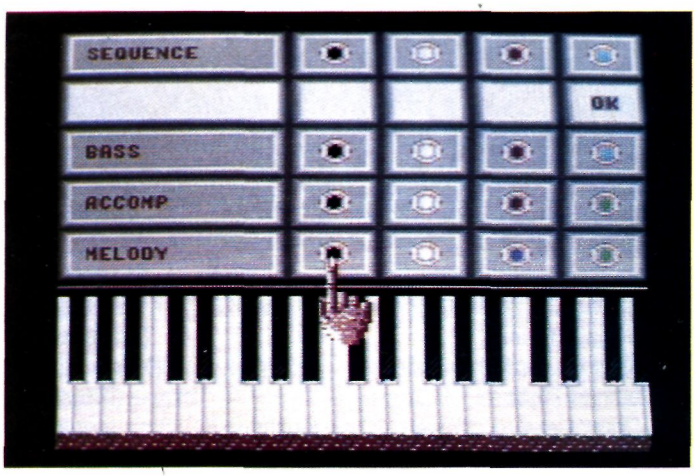
Auf diese Weise entsteht, je nachdem welche Inspirationen man gesammelt hat, ein interessantes Musikstück. Wenn man meint, einen Hit komponiert zu haben, steht man vom Klavier wieder auf.

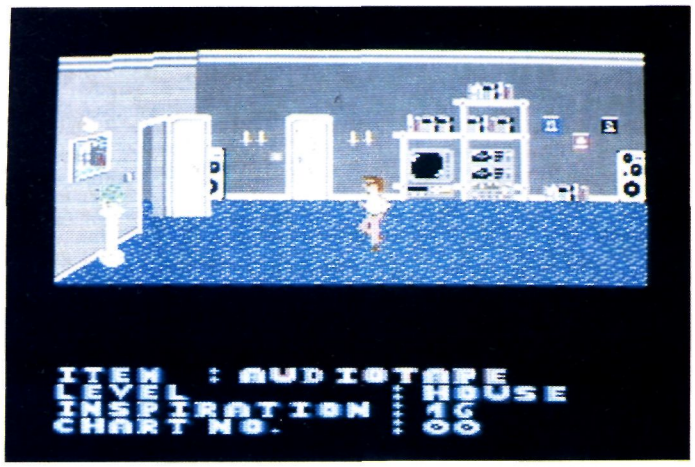
Ist man tatsächlich in der Top Ten vertreten, wird die Plazierung am unteren Bildschirmrand angezeigt.

Aber eine bloße Plazierung reicht dem Maestro nicht aus. Er weiß, daß er seine Komposition an einem Sythesizer stark verbessern kann.

Der beste Freund besitzt einen solchen Synthesizer. Wenn man dem Freund ein bestimmtes Item mitbringt, läßt einen dieser auch an seinen Synthy. Welches Item benötigt wird, wollen wir aber an dieser Stelle noch nicht verraten.

Am Synthesizer kann man die Sounds der verschiedenen Stim-





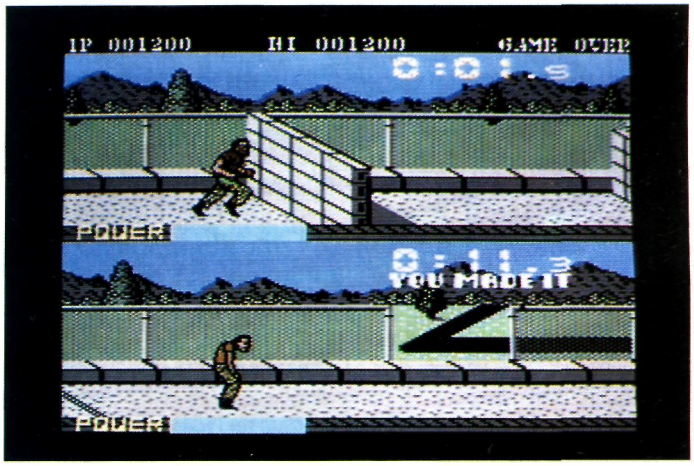

men noch einmal neu mischen. Je nachdem, wie gut dies gelingt, steigt oder fällt der Titel in der Hitparade.
Um der Komposition den letzten Schliff zu verpassen, müßte man in ein Tonstudio am Sound herumbasteln und noch Drumsequencen unterlegen. Das wird dem Musikfreak sofort klar.
Aber ohne ein Empfehlungsschreiben wird man jedoch wieder hochkantig aus dem Tonstudio herausgeworfen. Der Musikus muß sich von einer Person im Spiel ein Empfehlungsschreiben besorgen.
Wo und von wem wollen wir natürlich ebenfalls nicht verraten. Hat man sich jedenfall das Empfehlungsschreiben besorgt, kann man im Tomstudio seinen Sound noch mit Drums unterlegen.
Liegt man in der darauffolgenden Plazierung auf den ersten drei Plätzen, erhält der Musikus eine Einladung in eine beliebte Popsendung von 'Rainbow Arts TV'. In dieser Schlußsequenz muß man um die Gunst des Fernsehpublikums spielen. Ihr komponierter Sound wird eingespielt und dazu müssen Sie live am Keyboard spielen.
Bewegt man den Joystick nach rechts oder links, erklingen zwei Drums, drückt man ihn nach oben, klatscht der Star in die Hände, zieht man ihn nach unten, singt er einmal 'Nineteen'. Eine Skala am linken Bildschirmrand zeigt die Gunst des Publikums.
Taktgefühl und Präzision werden vom Spieler gefordert. Schlägt die Skala oben an, ist man automatisch auf Platz 1 der Hitparade. Um dem Erfolg ein Zeugnis zu geben, läuft ein niedliches Mädchen ins Fernsehstudio und überreicht dem musikalischen Genius eine goldene Schallplatte.
Ist einem diese Auszeichnung überreicht worden, hat man gewonnen, das Spiel ist vorbei. Nach jedem Spiel kann man sich seine Komposition anhören.
Mit den Zahlentasten 1-9 und 0 können alle Sounds in der Top Ten aufgerufen werden. 'To be on Top' ist ein hervorragendes Spiel, bei dem schnelle Joystickreaktionen, Kombinationsvermögen und Kreativität verlangt werden. Die Grafik ist sehr gut gelungen, der Sound, der in allen Situationen im Spiel verschieden ist, könnte gar nicht besser sein. Aufwendige Effekte erfreuen den Spieler an allen Stellen des Spiels.Aufgrund der intelligent gelösten Kompositionstechnik in 'To be on Top' gelingt es sofort, anhörliche Musikstücke zu komponieren.
Man muß eben nicht Mozart sein, um Spaß an 'To be on Top' zu haben.
**To be on Top ist top!**

| To Be On Top | |
|---|---|
| **System: C64** | |
| **Hersteller: Rainbow Arts** | |
| **Spaß/Motivation** | ★★★★★★★ 7 |
| **Grafik** | ★★★★★★★ 7 |
| **Sound** | ★★★★★★★ 7 |
| **Originalität/Idee** | ★★★★★★★ 7 |
| **Schwierigkeit** | ★★★★★★ 6 |
| **RCM-Wertung** | ★★★★★★★ 7 |

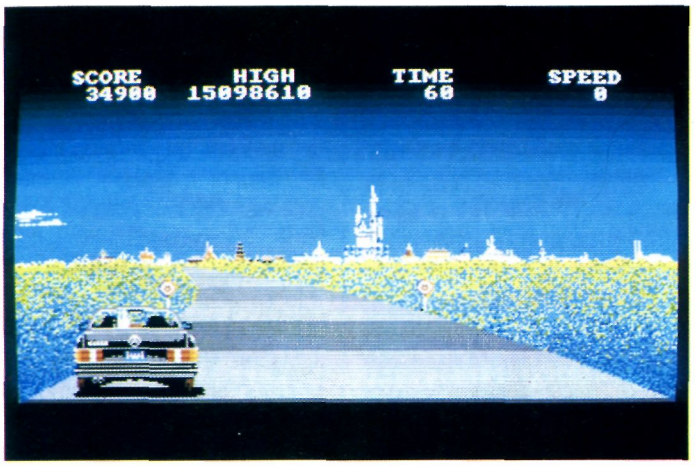
### Olympiade in der Kadettenschule

Mit Summergames von Epyx fing alles an. Fast jede namhafte Softwarefirma produzierte ein Olympiadespiel. Mal wird die Olympiade im schottischen Hochgebirke abgehalten (Highland Games), man im Wilden Westen (Western Games) oder in Californien (California Games).
Die Disziplinen sind in vielen Fällen gar nicht olympisch. Aber wen stört das? Hauptsache, man hat Spaß daran!
Recht lustig geht es auch bei Combat School von Ocean zu.
Als Absolvent einer Kadettenschule nehmen Sie an einem Härtetest teil, der für Ihre Karriere beim Militär von äußerster Wichtigkeit ist. Sieben verschiedene Disziplinen sorgen für eine Menge Abwechslung. Strengen Sie sich an, für jede Disziplin muß ein Mindestmaß an Punkten erkämpft werden, sonst werden Sie zum Schlaffi degradiert und aus ist der Traum, beim Militär Karriere zu machen.
In der ersten Disziplin können Sie schon zeigen, was in Ihnen steckt. Zwei Kadetten stehen an der Startlinie (einer davon sind Sie). In wenigen Sekunden wird der Startschuß gegeben. Es gilt, einen Hindernisparcour in einer vorgeschriebenen Zeit zu bewältigen.
Achtung! Auf die Plätze, fertig, los. Der Joystick muß hin- und hergerüttelt werden, um genügend Energie zum Laufen zu bekommen. Der Bildschirm ist gesplittet. Auf der unteren Bildschirmhälfte sehen Sie, wie der Gegner sich anstellt. Achtung, eine Mauer! Sie drücken den Feuerknopf und der Athlet springt mit einem Satz darüber .
Bei meinem Test zog ich mir einen Krampf in der Hand zu, mein Puls war auf 180, aber egal, ich hatte die vorgeschriebene Zeit unterboten.
Und so geht es weiter; diesmal mit einer ruhigeren Disziplin: Schießen. Mit Hilfe eines Fadenkreuzes und dem Feuerknopf müssen verschiedene Ziele getroffen werden, bevor es mit dem 'Iron Man Race' weitergeht, dem Rennen, das nur was für ganz harte Athleten ist.
Laufen, Rudern und schließlich hecheln Sie wieder auf Schusters Rappen über die Ziellinie.Schießen, Armdrücken, Schießen und eine Reckübung überstehen müssen Sie dann noch , dann steht Ihrer Karriere beim Militär nichts mehr im Wege.
Mit hat Combat School sehr viel Spaß gemacht. Ich bin sicher, auch Sie werden sich gut mit Combat School unterhalten. cb
**Bezugsquelle:**
**CSJ, Hannover**

| Combat School | |
|---|---|
| **System: C64** | |
| **Hersteller: Ocean** | |
| **Spaß/Motivation** | ★★★★★★★ 7 |
| **Grafik** | ★★★★★★★ 7 |
| **Sound** | ★★★★★★★ 7 |
| **Originalität/Idee** | ★★★★★ 5 |
| **Schwierigkeit** | ★★★★★★ 6 |
| **RCM-Wertung** | ★★★★★★ 6 |



# Brandheiße Spiele-Reviews




## Tolle Grafikdemo!

Mercedes, Porsche, Ferrari und Lamborghini stehen im Mittelpunkt eines neuen Autorennens von EH Ptroductions. In einem Kampf gegen die Zeit müssen Sie über amerikanische Highways rasen und in einer vorgeschriebenen Zeit die Ziellinie überschritten haben.

Mit einem Mercedes 560 SEC fängt alles an. Sie stehen mit Ihrem Vehikel an der Startlinie. Die Zeit läuft. Also Joystick nach vorne drücken und los geht's. Während Sie mit 257 Stundenkilometern Höchstgeschwindigkeit über den Highway donnern, läuft erbarmungslos eine Stoppuhr, die Ihnen angibt, wieviel Zeit noch bleibt, um ins Ziel zu rasen.

Sie heizen mit Ihrem Mercedes um die Kurven und überholen andere Fahrzeuge, wie zum Beispiel einen windschnittigen Porsche Turbo.

Wenn Sie durchs Ziel rauschen, bekommen Sie Bonuspunkte für die noch verbleibende Zeit und dann stehen Sie auch schon am Start für die nächste Highwayjagd. Diesmal sind kleine Hügel in die Rennstrecke eingebaut, die auf einigen Strecken für enorme Sprünge der Fahrzeuge sorgen, daß man Angst haben muß, die Federung bricht.

Haben Sie alle Rennstrecken gemeistert, gibt Ihnen der Computer die Möglichkeit, mit einem anderen Fahrzeug an den Start zu gehen. Grafisch ist Crazy Cars recht gut gelungen. Über den Sound kann man sich auch nicht beklagen. Ein digitalisierter Titelsound und digitalisierte Soundeffekte wie Motorengeräusche und Bremsenquietschen machen das Spiel sehr realistisch.

Einziger Schwachpunkt bei Crazy Cars liegt im Schwierigkeitsgrad: Wenn ein anderes Fahrzeug gerammt wird, explodiert der Mercedes nicht. Im Gegenteil, Ihr Gefährt fährt fast genauso schnell wie vorher weiter, als wenn nichts gewesen wäre.

Fazit: Crazy Cars ist eine tolle Grafikdemo für Liebhaber von schnittigen Autos.

Wer allerdings schwierige Autorennspiele bewältigen möchte, der ist mit anderen Simulationen wie Test Drive oder Grid Start besser beraten.

Hier zählen die Effekte, die kein anderes Autorennen zu bieten hat. cb

**Bezugsquelle:**
**DTM, Wiesbaden**

| Crazy Cars | |
|---|---|
| **System: Amiga** | |
| **Hersteller: Titus/EH-Prod.** | |
| **Spaß/Motivation** | ★★★★ 4 |
| **Grafik** | ★★★★★★★ 7 |
| **Sound** | ★★★★★★ 6 |
| **Originalität/Idee** | ★★★★ 4 |
| **Schwierigkeit** | ★ 1 |
| **RCM-Wertung** | ★★★★ 4 |

## Mit einem Buggy über Stock und Stein

Autorennen-Spiele scheinen im Moment ganz hoch im Kurs zu stehen.

Buggy Boy vom englischen Softwarehaus Elite ist das neueste Spiel dieses Genres für den C64. In diesem dramatischen Hindernisrennen steuert der Spieler keinen Formel-1-Wagen, sondern einen Buggy, wie er bei Autocrossrennen zu sehen ist.

Fünf verschiedene Strecken stehen dem Spieler zur Auswahl. Hat man einen Kurs ausgewählt, geht es auch schon los! Der Buggy steht an der Startlinie. Die Startampel zeigt Grün an, das wilde Rennen kann beginnen.

Mit dem Joystick betätigen Sie als 'Buggy-Bursche' die Gangschaltung. Mit lautem Motorengeräusch setzt sich das grafisch eindrucksvoll in Szene gesetzte Gefährt in Bewegung.

Sie müssen die Rennstrecke in einer vorgeschriebenen Zeit fahren. Dabei müssen verschiedene farbige Fähnchen eingesammelt werden.

Gar nicht so einfach, zumal auf der ganzen Rennstrecke Hindernisse verteilt sind. Felsbrocken, Bäume und sogar Mauern machen Ihnen das Rennfahrerdasein schwer.

Die Grafik ist bis auf den Buggy nicht besonders aufwendig und der Sound ist auch nicht der Rede wert.

Ein weiteres Manko ist die Tatsache, daß man es kaum schafft, die Rennstrecke in der vorgeschriebenen Zeit zu meistern. cb

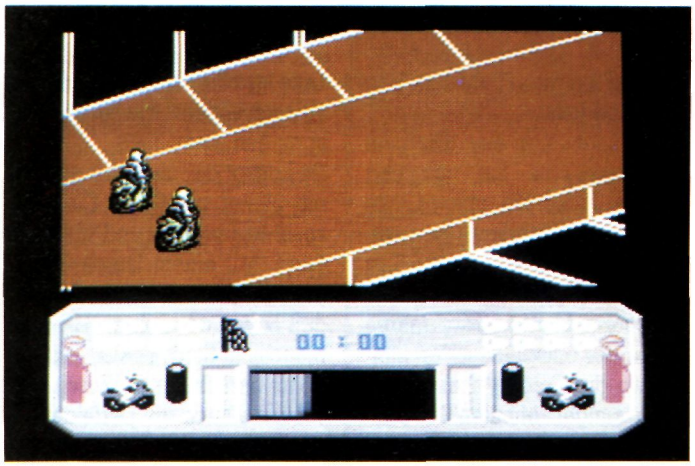
**Bezugsquelle:**
**Seemüller, München**

| Buggy Boy | |
|---|---|
| **System: C64** | |
| **Hersteller: Elite** | |
| **Spaß/Motivation** | ★★★★★ 5 |
| **Grafik** | ★★★★★ 5 |
| **Sound** | ★★★★★ 5 |
| **Originalität/Idee** | ★★★★★ 5 |
| **Schwierigkeit** | ★★★ 3 |
| **RCM-Wertung** | ★★★★★ 5 |

**Hotline**

☎ 05651-8778

**Jeden Donnerstag sind wir von 17.00 bis 19.00 Uhr für Sie am Telefon, um Ihre Fragen zu beantworten.**




## Tücken im Tunnel

In Mean Streak spielen Sie einen von der Gesellschaft unterdrückten Rebellen, der mit seinem Supermotorrad, dem Mean Streak, versuchen soll, durch einen Tunnel den Weg in die Freiheit zu finden.
Dieser Tunnel ist allerdings scharf bewacht. Ein Entkommen ist beinahe unmöglich.
Da stehen Mauern im Wege, Schlaglöcher sollen den Superhelden zum Stürzen bringen und das Allergemeinste ist, daß zwei andere Motorradfahrer versuchen, den Superhelden an der Flucht zu hindern.
Womit?
Natürlich mit einem Ballermann. Die Schüsse scheinen direkt vorne aus der Motorradlampe zu kommen.
Ebensoviel Tücke hat aber auch der Superheld zu bieten. Er ist nämlich, genau wie seine Widersacher, mit einem Knallermann bewaffnet.
Drückt der Spieler in Vertretung des Rebellen den Feuerknopf, wird ein böser Motorradfahrer nach dem anderen aus dem Sattel gehoben.
Mean Streak ist unheimlich rasant, wofür das perfekte Diagonalscrolling sorgt.
Über den Sound kann ich nur sagen: Vom Feinsten!
Das Spiel bietet übrigens die Möglichkeit mit zwei Motorradrebellen gleichzeitig zu spielen.
In der Zwei-Spieler-Option können sich beide Gegner im Tunnel gegenseitig abschießen. cb

**Bezugsquelle:**
**Compyshop,Mühlheim**

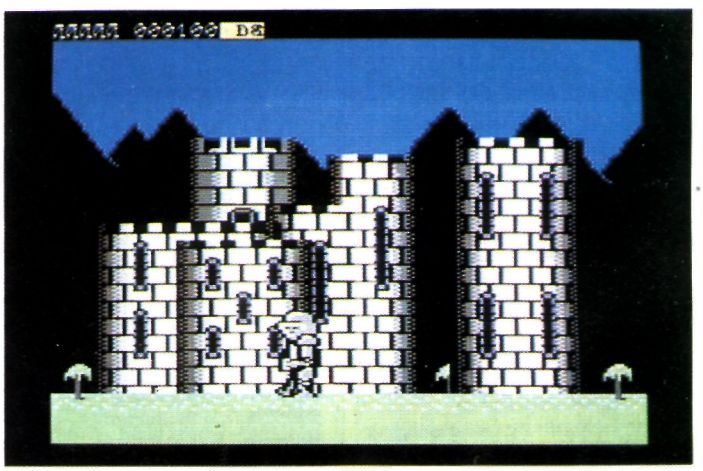
| Mean Streak | |
|---|---|
| **System: C64** | |
| **Hersteller: Mirrorsoft** | |
| **Spaß/Motivation** | ★★★★★★★ 7 |
| **Grafik** | ★★★★★★★ 7 |
| **Sound** | ★★★★★★★ 7 |
| **Originalität/Idee** | ★★★★★★ 6 |
| **Schwierigkeit** | ★★★★★★★ 7 |
| **RCM-Wertung** | ★★★★★★★ 7 |

## Aus Rampage wurde Ramparts

Go, das neue Label von U.S. Gold. Nach Trantor und Captain America bekommt der deutsche Spielefreak nun auch Go's neuestes Actionspiel mit Namen Ramparts zu Gesicht.
Zu Beginn des Tests möchte ich den interessierten Spielefreund auf eine Ähnlichkeit mit einem anderen, ebenfalls erst kürzlich erschienenen Programm vom Activision hinweisen.
Activision's Rampage beruht auf der selben, destruktiven Spielidee, tut dies aber auf lustigere Art und Weise.
Sir Griswold und Sir Larkin sind verflucht. Sie haben nur eines im Sinn: Zerstörung!
In England standen im Mittelalter eine Reihe von Burgen und Festungen in der Landschaft herum. Da die beiden verrückten Adeligen nichts besseres zu tun haben, klettern Sie auf Schloßmauern herum und schlagen alles platt, was Ihnen in den Weg kommt.
Die beiden Ritter sind erstaunlich groß für ihr Alter, wenn nicht sogar riesig (Ja, Mutationen gab es auch schon im Mittelalter). Das destruktive Schauspiel sehen sich die Bewohner der Gegend natürlich nicht mit an.
Hexen, feuerspeiende Drachen und ein paar Bauern, die ihrem Fürsten treu ergeben sind, geben den beiden Sirs Saures.
Steine, Felsbrocken und eine Menge Feuer sollen die beiden von der Arbeit abhalten.
Sie können Ramparts alleine spielen oder aber mit einem Mitspieler das Leben von vielen Burgbewohnern unsicher machen.
Sound und Grafik wissen zu überzeugen. Spaß macht das Spiel auch. Was will man mehr.
Wenn man einmal davon absieht, daß die Programmierer von Go sich ein wenig bei Activision's Rampage inspiriert haben, ist es ein sehr ordentliches Spiel. cb

**Bezugsquelle:**
**Elektronik von A-Z, Berlin**

| Ramparts | |
|---|---|
| **System: C64** | |
| **Hersteller: GO (US Gold)** | |
| **Spaß/Motivation** | ★★★★★★★ 7 |
| **Grafik** | ★★★★★★★ 7 |
| **Sound** | ★★★★★★★ 7 |
| **Originalität/Idee** | ★★★★★★ 6 |
| **Schwierigkeit** | ★★★★★★ 6 |
| **RCM-Wertung** | ★★★★★★★ 7 |



**Hotline**
☎ 05651-8778
**Jeden Donnerstag sind wir von 17.00 bis 19.00 Uhr für Sie am Telefon, um Ihre Fragen zu beantworten.**

# Skateboard-Fever

Mit Skate or Die können Sie die aufregende Welt der amerikanischen Skateboardfreaks entdekken. Rasante Action, ein bombastischer Titelsound und perfekte Animation machen Skate or Die von Electronic Arts sehr spielenswert.

Das Programmiererteam Michael Kosaka, David Bunch und Stephen Landrum brachte seine jahrelange Erfahrung in der Spieleprogrammierung, die es bei Projekten wie Summer Games, The Temple of Apshai Triology und GFL Football mitbekam, in die Programmierung des Top-Games Skate or Die ein.

Die drei Programmierer rasten wochenlang mit einem Skateboard durch die Korridore des Electronic Arts Gebäudes, um die Fahreigenschaften eines Skateboards besser umsetzen zu können. Daß sich die drei viel Mühe gemacht haben, zeigt Skate or Die.

Was Sie alles in Skate or Die erleben können, möchte ich Ihnen auf dieser Seite beschreiben:

Nachdem Sie mit Load 'EA',8,1 die Diskette geladen haben, erscheint auch schon das Titelbild. Ein digitalisiertes Musikstück mit Drums und anderen rockigen Instrumenten wird solange gespielt, bis Sie den Feuerknopf drücken.

Dann geht es ab ins Vergnügen. Skate or Die ist eine Art Olympiade für Skateboardfreaks. Bevor Sie sich aber an den witzigen Disziplinen erfreuen können, landen Sie erst einmal im Skate Shop.

Mit dem Cursor können Sie eine Reihe von Features auswählen. Da besteht die Möglichkeit, die Farbe des Skateboards zu ändern, sich die Highscores anzusehen oder sich für einen Wettkampf eintragen zu lassen.

Ein drolliger Punker, der im Skateshop als Verkäufer fungiert, lehnt über den Tresen und macht dumme Bemerkungen.

Als letztes müssen Sie im Skateshop entscheiden, ob Sie trainieren wollen oder ob Sie sich gleich in den Wettkampf wagen möchten. Sollte letzteres der Fall sein, werden Ihre im Wettkampf erzielten Ergebnisse auf Diskette gespeichert.

Das könnte, wenn Sie das erste mal spielen, recht blamabel ausfallen. Also wählen Sie zuerst 'Trainieren'.

Sie befinden sich nun auf dem Marktplatz. Von hier aus können Sie jede Skateboardarena bequem mit Ihrem Skater erreichen. Mit Graphity wurden die Disziplinen auf die Straße gesprüht.

Es gibt fünf verschiedene Disziplinen: Freestyle, Highjump, Jam, Race und Joust.

Sie können sich eine Disziplin auswählen, die Sie trainieren möchten oder Sie spielen alle fünf hintereinander. Ich entschied mich bei meinem Test zuerst für Freestyle.

Mit Ihrem Skateboarder fahren Sie zu der entsprechenden Straße, wo die Freestylebahn liegt. Nach einigen Minuten Ladezeit sieht man eine aufgeschnittene Röhre, auch Half Pipe genannt, auf dem Bildschirm.

Der Skater steht oben auf einem kleinen Vorsprung und wartet, bis der Wettkampf beginnt. Wenn Sie den Joystick bewegen, stürzt sich der Athlet in die Halbröhre. Sie können in der Röhre jetzt immer hin- und herfahren, um ordentlich Schwung zu holen. Wenn Sie genug Schwung haben, können Sie mit Ihren Kunststücken beginnen. Machen Sie doch einen Salto und landen wieder auf dem Skateboard.

Für alle tollen Kunststücke gibt es Punkte. Nach einer bestimmten Anzahl von Versuchen wird Ihre erreichte Punktzahl auf dem Bildschirm ausgegeben.

Danach befinden Sie sich wieder auf dem Marktplatz, wo Sie sich wieder eine Disziplin aussuchen können.

Diesmal nehmen Sie am besten Highjump. Im Prinzip ist der grafische Aufbau von Highjump genau derselbe wie bei Freestyle. Der einzige Unterschied besteht darin, daß am Ende der Bahn eine Latte installiert wurde, auf der Sie ablesen können, wie hoch Sie gesprungen sind.

Da kommt es schon mal vor, daß ein Skater nicht das Gleichgewicht halten kann und vom Brett fliegt. Mit einem lauten Scheppern fliegt er von seinem Brett auf den Boden der Röhre.

Manchmal fliegt er auch aus der Röhre heraus, vor die Zuschauertribüne. Durch die perfekte Animation sieht das unheimlich komisch aus. (Schadenfreude ist eben doch noch die beste Freude).

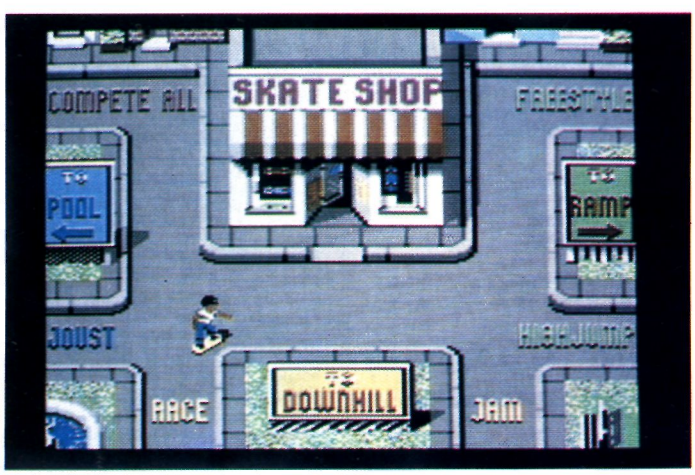
Nach fünf Versuchen im Skateboard-Hochsprung gelangen Sie wieder auf den Marktplatz. Als nächste Disziplin steht Pool Joust an. Hier befinden Sie sich in einer Arena. Mit einem Stock in der Hand müssen Sie dafür sorgen, daß Ihr Kontrahent, der sich ebenfalls mit seinen Skates herumtummelt, von den Brettern geholt wird.

Hauen Sie ihm doch den Stock einfach zwischen die Beine. Mit einer Rolle vorwärts fliegt der Gute von seinem Skateboard in die Arena. Nach einer gewissen Zeit erfolgt ein Wechsel und Ihr Gegner ist an der Reihe.

Nun müssen Sie fliehen und dem stöckchenschwingenden Skateboardfreak ausweichen.

Die letzten beiden Disziplinen sind heiße Skateboardrennen. In Downhill Jam rasen Sie mit einem Skateboard durch die Innenstadt. Die Geschwindigkeit macht schon ängstlich. Es geht nämlich die ganze Zeit bergab. Neben Ihnen rast ein anderer Skateboardfreak den Hang hinunter. Geben Sie Obbacht, daß er Sie nicht während der Fahrt mit seinen Ellenbogen von den Brettern stößt. Sie müssen der Ehre und der Punkte wegen schneller im Ziel ankommen als der Gegner.

Das Downhill Race, die letzte Disziplin ist wesentlich zivilisierter. Der Skateboardverein hat eine Strecke abgesteckt, über die Sie mit Ihrem Skateboard heizen müssen. Da gibt es Sprungschanzen, scharfe Kurven und viele an-



dere Hindernisse halten Sie in Atem.
Fazit: Skate or Die ist ein hervorragendes Actionspiel. Perfekt animierte Sprites, tolle Hintergrundgrafiken, digitalisierte Sounds von Rob Hubbard und fünf aufregende Disziplinen machen dieses wunderbare Spiel sehr empfehlenswert. cb

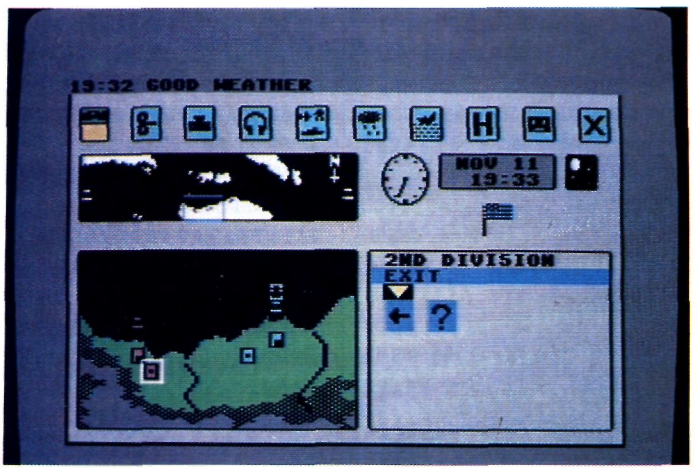
**Bezugsquelle:**
**CSJ, Hannover**

**Skate or Die**
**System: C64**
**Hersteller: Electronic Arts**

| | |
|---|---|
| Spaß/Motivation | ★★★★★★★ 7 |
| Grafik | ★★★★★★★ 7 |
| Sound | ★★★★★★★ 7 |
| Originalität/Idee | ★★★★★★★ 7 |
| Schwierigkeit | ★★★★★★ 6 |
| RCM-Wertung | ★★★★★★★ 7 |

## Krieg: Japan vs. Amerika

Activision präsentiert ein neues Strategiespiel für den C64, bei dem der Spieler die Schlacht zwischen amerikanischen und japanischen Streitkräften um eine Insel im Jahre 1942 nachspielen kann.
Sie haben als Spieler je nach Wahl die Kontrolle über die amerikanischen oder die japanischen Streitkräfte.
Guadacanal ist kinderleicht zu spielen. Sämtliche Aktionen werden mit Hilfe des Joysticks und einer intelligenten Iconsteuerung durchgeführt. Obwohl die Steuerung recht einfach zu bedienen ist, werden Sie eine ganze Weile brauchen, um den Computer zu schlagen.
Sie haben die Macht über die Luft- und Seestreitkräfte. Plazieren Sie Ihre Symbole strategisch günstig auf einer Karte. Transportschiffe, Häfen, Flüsse und andere örtliche Begebenheiten sollen Ihr Handeln beeinflussen.

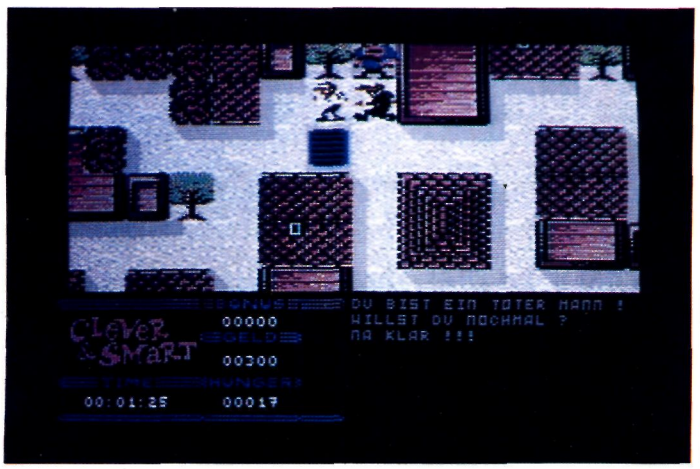



Grafisch ist Guadacanal nichts besonderes. Für ein Strategiespiel ist die Grafik allerdings zufriedenstellend. Sound ist so gut wie keiner vorhanden. Über Spielwitz und Motivation läßt sich natürlich streiten. Kann Kriegsspiel überhaupt witzig sein? Schwer zu beantworten, aber bei Guadacanal fließt wenigstens kein Blut. cb

**Bezugsquelle:**
**Elektronik von A - Z, Berlin**

**Guadacanal**
**System: C64**
**Hersteller: Activision**

| | |
|---|---|
| Spaß/Motivation | ★★★★★★ 6 |
| Grafik | ★★★★★★ 6 |
| Sound | ★★★ 3 |
| Originalität/Idee | ★★★★ 4 |
| Schwierigkeit | ★★★★★★★ 7 |
| RCM-Wertung | ★★★★★ 5 |




## Vom Comic-Helden zum Computersprite

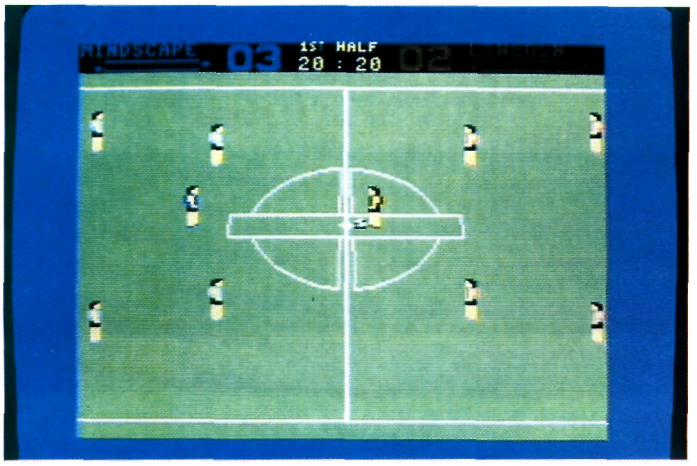
Clever and Smart heißt das neueste Spiel von Micro Partner, der Softwarefirma, die schon mit Titeln wie Mission Elevator, Werner und Western Games Furore gemacht hat. In Clever an Smart schlüpfen Sie in die Rolle der beiden tollpatschigen Agenten aus der gleichnamigen Comicserie. Sie können sich verkleiden, Schäden in der Kanalisation reparieren. Gesteuert werden die beiden Helden mit dem Joystick. Sie steuern beide Agenten gleichzeitig. Mit der Tastatur konnen Sie zwischen beiden hin- und herschalten. Haben Sie sich beispielsweise für Smart entschieden, trottet Clever brav hinterher. Die beiden sind einfach unzertrennlich. Sie haben einen Auftrag angenommen, der es in sich hat. Ein berühmter Geheimdienstwissenschaftler wurde entführt. Ihr Abenteuer haben Clever and Smart in einem Kindergarten, in der Kanalisation und auf einer Rennbahn zu bestehen. Comicgrafiken geben dem Spiel das letzte Flair. Clever and Smart ist ein verrücktes Spiel für verrückte Leute. cb

**Bezugsquelle:**
**CSJ, Hannover**

**Clever und Smart**
**System: C64/Amiga**
**Hersteller: Micro Partners**

| | |
|---|---|
| Spaß/Motivation | ★★★★★★★ 7 |
| Grafik | ★★★★★★★ 7 |
| Sound | ★★★★ 4 |
| Originalität/Idee | ★★★★★★★ 7 |
| Schwierigkeit | ★★★★★★ 6 |
| RCM-Wertung | ★★★★★★ 6 |

## Soccer par exellence




Als leidenschaftlicher Spieler von Commodore's Actionfußballspiel International Soccer und als Fan von Addictive's legendärem Strategiespiel Football Manager, habe ich mir schon lange ein Spiel gewünscht, daß beide Spielprinzipien kombiniert.
In Gary Lineker's Superstar Soccer werden dem Spieler sowohl strategische Entscheidungen in Trainergestalt als auch eine schnelle Hand im Actionteil des hervorragenden Soccerspieles abverlangt.
Die englische Firma Gremlin Graphics, die auf der Programmverpackung steht, ist nur der englische Lizensnehmer des Spiels. Programmiert wurde Superstar Soccer von der amerikanischen Firma Designstar Consultants, die duch Spiele wie Superstar Ping Pong und Indoor Sports bekannt geworden ist.
Erst im letzten Monat erschien Superstar Icehockey von Mindscape auf dem deutschen Markt (Roeske's Computer Mag. berichtete in der letzten Ausgabe). Für alle die, die Superstar Icehokkey schon kennen, kann ich sagen, daß Gary Lineker's Superstar Soccer über dasselbe Spielprinzip verfügt, wie Superstar Icehockey.
Für alle anderen möchte ich Superstar Soccer noch einmal ausführlich vorstellen:
Insgesamt verwaltet Superstar Soccer die Ergebnisse und Aktionen von 64 englischen Mannschaften, die in vier Ligen unterteilt sind. In der ersten Liga finden Sie so namhafte Mannschaften wie Arsenal London oder Manchester United.
Ihre Mannschaft befindet sich zu Beginn des Spiels noch unter den schlechtesten des Landes in der vierten Liga. Jede Liga, so auch die vierte besteht aus 16 Mannschaften; das bedeutet für Sie, daß Sie 15 Spiele pro Saison auszutragen haben.
Alle übrigen Spiele des englischen Fußballbundes, an denen Ihr Team nicht beteiligt ist, werden vom Computer intern ausgetragen und verwaltet.
Vor Beginn eines jeden Spiels können Sie sich in zahlreichen Menüs austoben. Sie können sich auf verschiedenen Tafeln ansehen, wieviele Spiele Ihr Team in der laufenden Saison noch spielen muß, wieviele Punkte Sie zur Zeit auf Ihrem Konto haben, wieviele Jahre Sie in der Liga gespielt haben,usw.
Sie werden es kaum glauben, aber Superstar Soccer merkt sich alle Ergebnisse der letzten 9 Saisons. Genau wie beim Deutschen Fußballbund wird auch ein englischer Pokalwettbewerb ausgetragen.
Sie können in einer scrollenden Tabelle alle Ergebnisse begutachten. Jede, der vom Programm verwalteten Mannschaften, hat eine andere Spielstärke, so daß jedes Spiel auf's Neue immer eine Herausforderung bleibt, zumal sich die Mannschaften im Laufe der Zeit auch verbessern oder verschlechtern können (ganz wie im richtigen Leben).
Wenn man vom Begutachten von Tafeln absieht, können Sie auch vor dem Spiel einige Aktionen tätigen; so ist es zum Beispiel möglich, die Namen der Spieler und Ihrer Mannschaft an heimische Gefilde anzupassen. Sie können Spieler aus Ihrer Mannschaft auf dem nationalen Spielermarkt kaufen und verkaufen.
Sie können ebenfalls Ihre Mannschaft ins Trainingslager schikken oder die Mannschaftsaufstellung verändern. Alle Aktionen erfolgen über Joystick. Wenn Sie alles zu Ihrer Zufriedenheit eingestellt haben, kommen Sie in ein Menü, das die Features für das nächste Spiel festsetzt.
Länge der Spielzeit und Trikotfarbe können eingestellt werden. Wenn alle Parameter Ihren Wünschen entsprechen, müssen Sie in diesem Menü noch drei Rollen besetzen: Den Torwart, den Mittelstürmer und den Trainer, der während des Spiels taktische Anweisungen gibt. Es muß eingestellt werden, ob die Position vom Computer, von Port 1 oder Port 2 gesteuert werden soll. Ich ließ bei meiner Mannschaft den Computer die Rolle des Trainers und des Torwarts übernehmen.
Nach einer Ladezeit beginnt auch schon das erste Spiel. Ich steuerte den Mittelstürmer, während der Computer alle anderen Rollen besetzte. Nach Spielunterbrechungen wie Freistößen oder Einwürfen, wird das Spiel für kurze Zeit unterbrochen und der Trainer gibt letzte Anweisungen.
Das Spiel ist sehr rasant. Man kann gute Kombinationen spielen, tolle Pässe schlagen und Tore schießen. Ein Schiedsrichter ahndet Fouls und andere Regelverstöße.
Superstar Soccer ist ein wahnsinnig gutes Fußballspiel, das sowohl strategische als auch Actionelemente in sich vereinigt.
Einziger Schwachpunkt ist die wenig aufwendige Grafik und der schwache Sound. Wem es aber nicht nur auf Äußerlichkeiten ankommt, der ist mit Gary Lineker's Superstar Soccer bestens bedient. cb

**Bezugsquelle:**
**Compy Shop, Mühlheim**

| **Superstar Soccer** | |
|---|---|
| **System: C64** | |
| **Hersteller: Gremlin Graphics** | |
| **Spaß/Motivation** | ★★★★★★★ 7 |
| **Grafik** | ★★★★ 5 |
| **Sound** | ★★★ 3 |
| **Originalität/Idee** | ★★★★★★★ 7 |
| **Schwierigkeit** | ★★★★★★ 6 |
| **RCM-Wertung** | ★★★★★★ 6 |


# Kennen Sie schon unseren Disketten-Service?

**Auf der Seite 79 erfahren Sie alles wissenswerte über die Programme, die in diesem Monat auf den Service-Disketten vorhanden sind. Publik-Domain-Programme und die Programme aus diesem Heft bieten interessante Software zu Niedrigstpreisen.**
**So erhalten Sie für wenig Geld eine umfangreiche Programmsammlung mit Anwender- und Spielprogrammen.**

## Die olympischen Winterspiele 1988

1988 finden in Kanada, genauer gesagt, in Calgary die offiziellen, fünfzehnten olympischen Winterspiele statt. Rechtzeitig zu diesem Weltereignis bietet Tynesoft die olympischen Winterspiele für den C64 an.

Winter Olympiad 88 heißt der Sportspielspaß mit den hervorragenden Grafiken, die besser gelungen sind als bei Wintergames von Epyx. Wintergames war wohl das Vorbild von Tynesofts neuestem Sportspiel, das sieht man ohne Zweifel.

Bis zu sechs Spieler können an dieser Olympiade teilnehmen. Jeder Spieler kann seinen Namen eingeben und sich eine Nation aussuchen, für die er an den Start gehen möchte. Nach jeder Disziplin wird die Nationalhymne des Gewinners gespielt.

Der interessierte Spielerfreund wird merken, wie krass die Parallelen zu den Wintergames sind. Sie gelangen jetzt in Optionsmenü 3, in dem Sie sich einzelne Disziplinen für einen Wettkampf herauspicken können. Es ist natürlich auch möglich, alle Disziplinen hintereinander zu spielen.

An dieser Stelle möchte ich mich einmal über die Kassettenversion von Winter Olympiad 88 beschweren und Ihnen von einem Kauf der Kassettenversion abraten. Wegen der aufwendigen Grafiken dauert es sehr lange, das Menü in den Computer zu laden.

Aber nicht nur das. Jede einzelne Disziplin muß nachgeladen werden, so daß Sie sehr lange Wartezeiten in Kauf nehmen müssen, bis Sie endlich starten können. Es dauert schon einige Minuten, bis Sie sich die Eröffnungszeremonie ansehen können, die ebenfalls aus Wintergames abgekupfert wurde.

Wenn Sie sich die Eröffnungszeremonie angesehen haben, können Sie sich erneut auf eine lange Wartezeit einstellen. Am besten, Sie lesen in der Zwischenzeit ein gutes Buch oder kochen sich einen Kaffee.

Die erste Disziplin heißt Schanzensprung. Man sieht den Athleten oben auf der Schanze stehen. Drücken Sie den Feuerknopf und er stürzt die Schanze herunter. Das Szenario wechselt für kurze Zeit. Es besteht nun die Möglichkeit, die Skier richtig für eine gute Landung auszurichten.

Wenn Sie gut gelandet sind, bekommen Sie Punkte für die Sprungweite und für den Stil der Landung. Wenn Sie sich beim Sprung nicht die Gräten gebrochen haben, geht es weiter mit der nächsten Disziplin, dem Abfahrtslauf.

Sie heizen einen Hang herunter und versuchen, möglichst schnell über die Piste zu fliegen.

Eine noch größere Herausforderung als der Abfahrtslauf ist der Biathlon. Durch Rechts- und Linksbewegungen mit dem Joystick bewegt sich der Athlet vorwärts. Wenn Sie eine Geländestrecke zurückgelegt haben, kommen Sie in einen Schießstand, wo Sie fünf Ziele mit fünf Patronentreffen müssen. Schießen Sie bloß nicht vorbei, das gibt nämlich Punktabzüge.

Nach dem Schießen müssen Sie noch eine Geländestrecke zurücklegen, bis Sie im Ziel ankommen. Beim Slalom müssen Sie Ihren Athleten zwischen Fahnenstangen in Slalommanier durchbugsieren. Bei der letzten Disziplin wurde ich schon ziemlich wütend.

Das Bobrennen aus Wintergrames wurde beinahe 1:1 kopiert. Sie sitzen in einem Bob, pesen die Piste herunter. Auf die Strekkenübersichtskarte sollten Sie genau achten. Sie zeigt, wo die Gefahrenstellen auf der Piste sind. Also, nicht aus der Bahn fliegen!

Die Grafik und der Sound sind besser als bei Wintergames. Ich muß aber deutlich machen, daß Tynesoft fast alle Features von Wintergames abgekupfert hat. Schade, denn der Grafik und dem Sound entnehmend, kann man sehen, daß es sich um recht fähige Programmierer handelt. cb

**Bezugsquelle:**
**Electronik von A-Z, Berlin**

| Winter Olympiad 1988 | |
|---|---|
| **System:** | **C64** |
| **Hersteller:** | **Tynesoft** |
| **Spaß/Motivation** | ★★★★★ 5 |
| **Grafik** | ★★★★★★★ 7 |
| **Sound** | ★★★★★★★ 7 |
| **Originalität/Idee** | ★ 1 |
| **Schwierigkeit** | ★★★★★ 5 |
| **RCM-Wertung** | ★★★★★ 5 |

## Stupide Ballerei

Action Force heißt ein neues Actionspiel aus dem englischen Softwarehaus Virgin, das nach einer in England bekannten Spielzeugserie entstanden ist.

Sie steuern in diesem rasanten Actionspiel einen Hubschrauber oder in manchen Situationen auch einen Rocketman, die beide das gleiche Ziel haben: Soviele Feinde abballern, wie nur irgend möglich.

Action Force hat eine wunderhübsche Grafik; manchmal ist auch strategische Arbeit zu leisten. So muß man beispielsweise Brücken zerstören, damit der Feind keinen Nachschub nachliefern kann.

Sie können auch einmal kurz landen und neue Munition oder Treibstoff aufnehmen. Trotz dieser netten Features ist und bleibt Actionforce ein stupides Ballerspiel.

Die Grafik ist über dem Durchschnitt, ansonsten lockt es aber keinen Spielefreak hinter den Joystick hervor. cb

| Action Force | |
|---|---|
| **System:** | **C64** |
| **Hersteller:** | **Virgin Games** |
| **Spaß/Motivation** | ★★★ 3 |
| **Grafik** | ★★★★★★ 6 |
| **Sound** | ★★★★★ 5 |
| **Originalität/Idee** | ★★★★ 4 |
| **Schwierigkeit** | ★★★★★★ 6 |
| **RCM-Wertung** | ★★★★★ 5 |

# Ein Spiel wie eine Droge

Beinahe wäre dieser Spieleteil nicht rechtzeitig fertig geworden. Die Softwarefirma Time Warp Productions wäre Schuld gewesen.
Die Jungs aus Gütersloh schickten mir einen Tag vorm Redaktionsschluß noch ein neues Game mit dem Namen Great Giana Sisters.
Ich spielte und spielte und vergaß dabei die Zeit und die Texte, die ich noch abgeben mußte. Zwei Stunden spielte ich Great Giana Sisters, bis der Chefredakteur energisch seine Texte forderte.
Ich hatte während des Spiels jedes Zeitgefühl vergessen, denn ich befand mit in einer anderen Welt, in der Traumwelt von der kleinen Giana.
Eines Nachts schlief das kleine entzückende Mädchen tief und fest und wie alle Kinder in diesem Alter träumte sie von vielen schönen Dingen. Normalerweise träumen die Menschen scheinbar zusammenhanglosen Blödsinn.
Was die kleine Giana aber geträumt hat, ist Ihnen bestimmt noch nicht passiert. Giana befindet sich in ihrem Traum in einer fremden unheimlichen Welt, voller Fantasie.
Alles scheint dort anders zu sein, als in der Welt, die wir kennen. Unheimliche Kreaturen, Gebilde und Gebäude befinden sich in dieser Welt. Giana fühlt sich frei wie ein Vogel. Die Grenzen der Schwerkraft sind so gut wie aufgehoben.
Giana fühlt sich zwar wohl in dieser Traumwelt, doch möchte sie in die richtige Welt zurückkehren. Das geht aber nicht so einfach. Dazu braucht sie den magischen Riesendiamanten und dieser ist irgendwo in der Traumwelt versteckt.
Aber wo? das ist die Frage. Aus dem wunderbaren Traum wurde Realität. Giana läuft in der Traumwelt herum und sucht den Wunderdiamanten.
Soweit die Hintergrundstory zu Great Giana Sisters, die ich vor dem Spiel durchlas. Nach dem Laden erschien das Titelbild auf meinem Monitor. Ein fünfstimmiger Sound weckte mich aus meinen Träumen.

Digitalisierte Drums und eine schöne Melodie machen dieses Musikstück zum Hörerlebnis. Nachdem ich mir den Sound ein paar Mal angehört hatte, drückte ich Space, so daß das Spiel automatisch eingeladen wurde.
Great Giana Sisters besteht aus 32 Levels. Aufgrund der Tatsache, daß alle Levels im Speicher sind, muß der Computer nicht nachladen und auf diese Weise wird der Spielfluß auch nicht gestört.
Great Giana Sisters kann mit ein oder zwei Spielern gespielt werden. Wenn Giana ein Leben verloren hat, kann ihre Schwester Maria mitträumen.
Gesteuert wird natürlich mit dem Joystick. Die Traumwelt ist, wie schon gesagt, in 32 Levels unterteilt. Giana kann überall herumlaufen und springen. Natürlich gibt es auch Gefahren bei diesem Kletterspielspaß. Giana darf nicht in Abgründe fallen oder Kollisionen mit Monstern haben, sonst verliert sie eins ihrer fünf Bildschirmleben.
Das ist sehr schlecht, denn wenn Giana ihre Aufgabe nicht erfüllen kann, muß sie ewig in der Traumwelt bleiben. Das Wuinder des Extralebens gibt es auch in Great Giana Sisters. In jedem Level sind Kristalle auf den Felsen oder in der Luft. Giana braucht nur 100 Kristalle einzusammeln, um ein Extraleben zu bekommen.
Auf dem Weg durch die Traumwelt wird Giana viele Gesteinsbrocken finden, die auf den ersten Blick nutzlos erscheinen. Man sollte unter die Brocken springen. Manchmal lösen sich weitere Kristalle oder Bonussymbole, die für den weiteren Erfolg im Spiel von besonderer Wichtigkeit sein können.
Wenn Giana die Symbole berührt, kann sie die Bonussymbole benutzen. Da gibt es beispielsweise ein Feuerrad, durch das sich Giana plötzlich in eine Punkerin verwandelt. Ihre blonden Haare stehen jetzt zu Berge und das nette Mädchen strotz nur so vor Kraft. Sie kann jetzt Monster zerquetschen, wenn sie von oben auf die Monster springt. Giana hat jetzt sogar Kraft, Steine zu zerschlagen.
Ein weiteres Symbol ist der Blitz. Mit diesem Symbol kann Giana Traumblasen werfen, durch die alle Monster abgeschossen werden können. Wenn Giana einen Doppelblitz findet, schießt sie Traumblasen, die an allen Hindernissen abprallen und sich so selbständig ihren Weg suchen.
Wenn Giana eine Erdbeere findet, dringen die Traumblasen durch manches Gestein hindurch. Findet Giana einen Lolli, bekommt sie ein Extraleben. Nach dem Fund einer Uhr versetzt Giana alle Feinde in einen Zauberschlaf. Wenn sie eine Zauberbombe findet, lösen sich alle Feinde auf Druck der Space Taste in Luft auf.
Es gibt noch viele andere Überraschungen in Great Giana Sisters, die wir aber nicht alle verraten möchten. Wenn Giana alle fünf Leben verloren hat, haben Sie die Möglichkeit, Ihren Score auf der Diskette abzuspeichern. Wenn das mal kein Anreiz ist.
Fazit: Great Giana Sisters ist ein sehr interessantes Spiel mit 32 Levels, die aller voller Spielspaß stecken. Der Sound könnte nicht besser sein. Ist ja auch kein Wunder, Soundmagier Chris Hülsbeck war wieder am Werke.
Die Grafik ist zwar nicht besonders aufwendig, aber durchaus angemessen. Ich bin von Great Giana Sisters begeistert und ich wette, Sie werden es auch sein. Ein aufregenderes Spiel kann man gar nicht kreieren. Das Spielprinzip ist zwar vom Spielautomaten Super Mario Brothers übernommen worden. Time Warp Production holt Ihnen den Supertitel in Ihr Wohnzimmer.
Wäre ich Arzt, würde ich Ihnen vom Kauf des Spiels abraten, denn es macht süchtig. Als Spielfreak sage ich: Wenn Sie sich Great Giana Sisters zulegen, werden Sie bestimmt genauso viel Spaß damit haben wie ich. cb

**Vertrieb: Rushware**

**Great Giana Sisters**
**System: C64 (Amiga i.V.)**
**Hersteller: Time Warp Prod.**

| | |
|---|---|
| **Spaß/Motivation** | ★★★★★★★ **7** |
| **Grafik** | ★★★★★★ **6** |
| **Sound** | ★★★★★★★ **7** |
| **Originalität/Idee** | ★★★★★★★ **7** |
| **Schwierigkeit** | ★★★★★★ **6** |
| **RCM-Wertung** | ★★★★★★★ **7** |

## Von Schurken und Helden

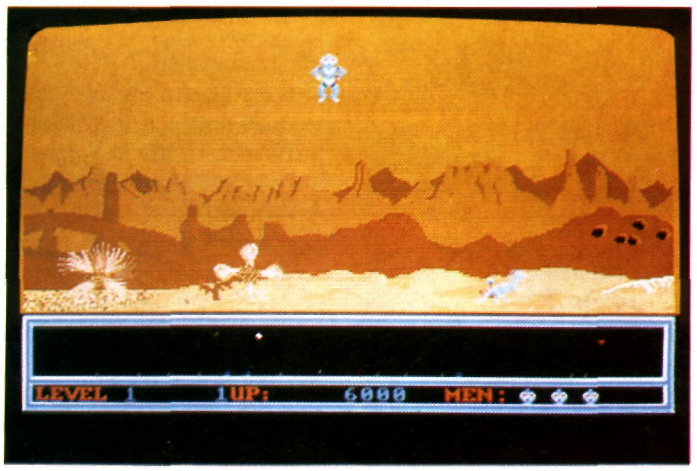
Vor etwa einem Jahr erschien Ninja Mission für nur DM 14,95 in Mastertronics C64-Low-Budgetreihe MAD (Mastertronics added Dimension).
Nachdem der Amiga auch in England ganz groß im Kommen ist, hat sich Mastertronic entschlossen, auch Low Budgetspiele für den Amiga zu veröffentlichen. Der einfachste Weg möglichst schnell neue Spiele zu produzieren, bestand darin, einfach einen guten C64 titel zu konvertieren. So geschehen mit Ninja Mission: Akuma, ein mieser Schurke hat es gewagt, sieben wertvolle Jaden aus einem kleinen Dorf mit Namen Tambo Machi zu stehlen, und das, obwohl sie gut versteckt im Dorf aufbewahrt wurden.
Akuma hat sich die wertvollen Jaden, die der ganze Stolz des Dorfes waren, einfach unter die Arme geklemmt und hat sie in seinen Palast geschleppt.
Ein Ninja, mit fernöstlichen Kampfsportarten wie Karate, Judo und Aikido bestens vertraut, bekommt von der Dorfbevölkerung den Auftrag, die Jaden zurückzuholen.
Die Rolle des Ninjas wurde natürlich Ihnen zugedacht. Daß Ihre Aufgabe nicht leicht ist, wird Ihnen klar, als Sie den Palast von außen sehen: Überall stehen Wachen!
Akuma hat sein zusammengeklautes Vermögen in bestens ausgebildete Wächter investiert. Sobald Sie den Palast betreten, gehen die Probleme auch schon los. Ein wildgewordener Karatefreak stürmt auf Sie zu und will Ihnen den Hintern versohlen.
Doch das lassen Sie sich nicht gefallen. Sie könnten dem Wächter mit ein paar Karatehieben zeigen, wer der Herr im Hause ist, aber Sie wollen sich ja die Hände nicht schmutzig machen. Glücklicherweise ist der Ninja mit einer Reihe von Kampfsternen und Messern ausgerüstet, die Sie den Wachen munter entgegenwerfen können.
Wenn Sie sich aber doch schmutzig machen wollen, können Sie auch im Nahkampf den Wächter besiegen.
Gesteuert werden sämtliche Aktonen mit dem Joystick.
Den Palast, der aus fünf Etagen besteht, gilt es zu durchsuchen. Im untersten Stockwerk werden Sie noch keine Probleme haben, sich der Wächter zu entledigen. Aber schon in der zweiten Etage greifen Sie manchmal zwei Wächter gleichzeitig an.
Wenn Sie in einer Etage alle Jaden entdeckt haben, besteht die Möglichkeit, über eine Luke in der Decke das nächste Stockwerk zu erreichen.
Ihr Ninja und jeder Wächter verfügen über einen Energiestreifen, der sich am oberen Bildschirmrand befindet. Durch die verschiedenen Kampftechniken muß die Energie des Wächters geschmälert werden. Dies erfolgt am leichtesten über die schon angesprochenen Kampfsterne und Wurfmesser.
Die Grafik ist bei Ninja Mission sehr detailliert. Hervorragende Hintergrundgrafiken, eine perfekte Animation des Ninjas und der Wächter machen Ninja Mission sehenswert. Hörenswert wird Ninja Mission durch den hervorragenden Sound. Asiatische Melodien stimmen den Spieler in die Spielhandlung ein. Schade nur, daß keine digitalisierten Kampfschreie zu hören sind. Durch ein solches, zusätzliches Feature wäre Ninja Mission noch realistischer.
Aufgrund der intelligenten Joysticksteuerung und der nervenaufreibenden Handlung ist Ninja Mission auch sehr spielenswert. Ninja Mission ist sein Geld wert! cb

**Bezugquellen:**
**Soyka Datentechnik, Bochum**
**Intersoft LaSch, Oberhausen**
**Elektronik von A-Z, Berlin**
**Profisoft**

| **Ninja Mission** | |
|---|---|
| **System: Amiga** | |
| **Hersteller: Mastertronic** | |
| **Spaß/Motivation** | ★★★★★★★ 7 |
| **Grafik** | ★★★★★★★ 7 |
| **Sound** | ★★★★★★★ 7 |
| **Originalität/Idee** | ★★★★★★ 6 |
| **Schwierigkeit** | ★★★★★★ 6 |
| **RCM-Wertung** | ★★★★★★★ 7 |

## Helden im Weltraum

Die Bewohner des Planeten Zyphoor feiern gerade ein großes Fest. Als die Sektkorken knallen, greifen die bösen Aliens an.
Bei dem Angriff selbst wird zwar niemand getötet, die Aliens nehmen aber Geiseln, die dann auf dem Heimatplaneten genüßlich zu Tode gefoltert werden.
Da kommen Sie gerade richtig. Sie müssen den Frevel, mit einer Laserpistole und einem Bündel Netze bewaffnet, verhindern.
Ähnlich wie bei Dropzone auf dem C64 fliegen Sie, der Space Ranger über eine Planetenoberfläche, auf der man auch landen kann. Auf einem Radarbildschirm wird die Position der einzelnen Planetenbewohner angezeigt. Sie können diese folgendermaßen retten: Zuerst fliegen Sie mit Hilfe des Joysticks an sie heran, dann drücken Sie den linken Mousebutton, ein Netz wird über die Wesen ausgeworfen, das dann in den Weltraum aufsteigt.
Ein Wahnsinnsspaß mit filmartiger Animation und einem überdurchschnittlichen Sound. cb




| **Space Ranger** | |
|---|---|
| **System: Amiga** | |
| **Hersteller: Mastertronic** | |
| **Spaß/Motivation** | ★★★★★★★ 7 |
| **Grafik** | ★★★★★★★ 7 |
| **Sound** | ★★★★★★ 6 |
| **Originalität/Idee** | ★★★★★★★ 7 |
| **Schwierigkeit** | ★★★★★★ 6 |
| **RCM-Wertung** | ★★★★★★★ 7 |

# Hokus Pokus

Vor langer, langer Zeit lebten im alten England zwei Brüder mit Namen Learic und Leanoric, die sich beide ganz gut in der Zauberkunst verstanden.
Den lieben langen Tag probten sie Ihre Zaubersprüche nicht weit von ihrer Hütte, die sie gemeinsam bewohnten. Rings um die Hütte war nämlich ein großes Waldstück, in dem sie mit ihren Zaubereien niemand stören konnten.
Ihr Leben hätte so harmonisch verlaufen können, wenn es nicht eines Tages zu einem fürchterlichen Streit gekommen wäre. Learic behauptete nämlich: 'Ich bin der beste und mächtigste Zauberer im ganzen Land!' 'Das ist ja gar nicht wahr, Du alter Lügner' konterte Leanoric. Vom Haß geprägt, kam in den beiden Brüdern der Gedanke auf, sich gegenseitig umzubringen.
Nur wie? Das war die große Frage. Da die beiden Greise schon zu alt waren, mit Messern, Pistolen, Knüppeln oder Karatetritten aufeinander loszugehen, wählten sie eine Möglichkeit, die ihnen angemessen erschien: Sie wollten sich mit Zaubersprüchen bekämpfen, solange, bis einer der beiden Brüder tot umfällt.
Die Rolle von Learic wird in diesem Kampf Ihnen zugedacht. Ihre Aufgabe besteht darin, zwölf Zutaten für einen Zauberbrei zu finden. Mit Hilfe des Zauberbreis fällt Ihnen dann der vernichtende Zauberspruch ein. Also, ran an die Arbeit.
Sie steuern Learic durch Wälder, Wiesen und das Dorf, um nach den wichtigsten Kräutern zu suchen.
Aber aufgepaßt! Ihr Bruder Leanoric wuselt ebenfalls im Wald herum. Er ist nämlich, genau wie Sie, auf der Suche nach Zutaten. Nehmen Sie die Beine in die Hand und beeilen Sie sich. Leanoric darf die Zutaten auf keinen Fall eher finden als Sie. Sonst ist es aus mit Ihrem Zaubererdasein. Mit Hilfe eines Kompasses, der sich am rechten Bildschirmrand befindet, können Sie feststellen, wo Leanoric herumfleucht. Wenn Learic alle 12 Zutaten in Form von Kräuterpflanzen eingesammelt hat, gilt es wieder, den Hexenkessel zu finden, an dem Sie gestartet waren.
Hier müssen Sie alle Kräuter hereinwerfen, einmal kräftig umrühren und dann ist es soweit. Der Zauberbrei und damit der Zauberspruch ist fertig. Mit Hilfe eines Zauberbuchs, das sich am linken Bildschirmrand befindet, ordnet man den verschiedenen Elementen des Zauberspruches bestimmte Funktionen zu. So ist es beispielsweise möglich, das Brüderchen in einen Zombie zu verwandeln.
Ist der Zauberbrei fertig, müssen Sie immer der Kompassnadel folgen und so Ihren Bruder Leanoric finden. Haben Sie ihn gefunden, geht der Rabatz los: Blitze fliegen von einem Zauberer zum anderen, solange, bis einer von beiden Brüdern tot umfällt.
Hat der computergesteuerte Kontrahent bessere Kräuter gesammelt als man selber, heißt es Abschied nehmen von der guten alten Erde.
Die Grafik von Mastertronics Feud ist sehr detailreich und farbenfroh wie aus einem Bilderbuch. Für Abwechslung ist ebenfalls gesorgt. Es gilt, eine riesengroße Landschaft nach Kräutern abzusuchen. Man wird gar nicht alle Bilder finden können, weil das Gebiet so riesig ist.
Man entdeckt unzählige neue Dinge, wie verlassene Hütten, Brücken, Wasserläufe oder Zauberbäume, unter denen besonders viele Kräuter stehen.
So gut die Grafik und die Handlung auch sind, so schlecht ist der Sound. Bis auf ein paar Pieptöne hat Feud in dieser Beziehung nichts zu bieten. cb

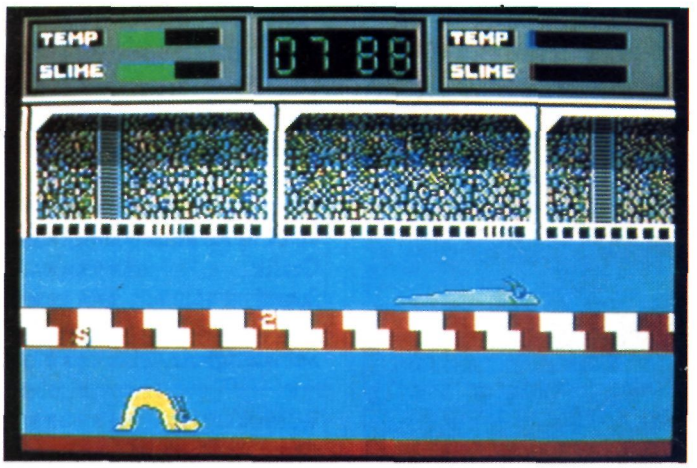
**Feud**
**System: C64/Amiga**
**Hersteller: Mastertronic**

| | |
|---|---|
| Spaß/Motivation | ★★★★★★ 6 |
| Grafik | ★★★★★★★ 7 |
| Sound | ★★ 2 |
| Originalität/Idee | ★★★★★★★ 7 |
| Schwierigkeit | ★★★★★★ 6 |
| RCM-Wertung | ★★★★★★ 6 |

# Originell

**aber grafisch schwach**

Fast jede namhafte Softwarefirma möchte mit einem mehr oder weniger guten Sportspiel an den Erfolg von Epyx Games Serie anknüpfen. So bietet Activision ein neues Olympiaspiel, das den Spieler auf einen fernen Planeten ins Universum versetzt.



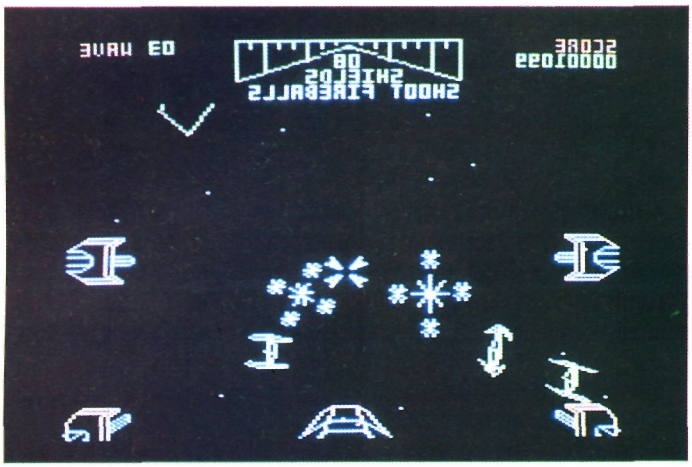

Hier stehen nicht Disziplinen wie Skiabfahrtslauf oder Bobrennen auf der Tagesordnung, nein, hier geht es um ganz andere Dinge. Kopfschleudern, Parapsychologisches Judo, Verwandlungsmarathon, Weltraumhockey und ein Wurmwettrennen.
Alle Disziplinen sind originell, das muß man Activision lassen. Leider ist die Grafik im Vergleich zu anderen Spielen dieses Genres sehr schlecht.
Beim Weltraumhockey bekommt der Spieler nichts anderes als ein viereckiges Feld, das in alle vier Himmelsrichtungen scrollt und drei Bälle zu Gesicht. Zwei Bälle müssen versuchen, einen dritten Ball in eins der Tore zu schießen. Das ganze ist recht witzig, aber über die Grafik kann kein Mensch lachen. Schade, aus diesen originellen Disziplinen hätte man mehr machen können. cb

**Vertrieb: Ariolasoft**

**Galactic Games**
**System: C64**
**Hersteller: Activision**

| | |
|---|---|
| Spaß/Motivation | ★★★★ 4 |
| Grafik | ★★★ 3 |
| Sound | ★★★★ 4 |
| Originalität/Idee | ★★★★★★★ 7 |
| Schwierigkeit | ★★★★★ 5 |
| RCM-Wertung | ★★★★★ 5 |

# Krieg der Sterne I



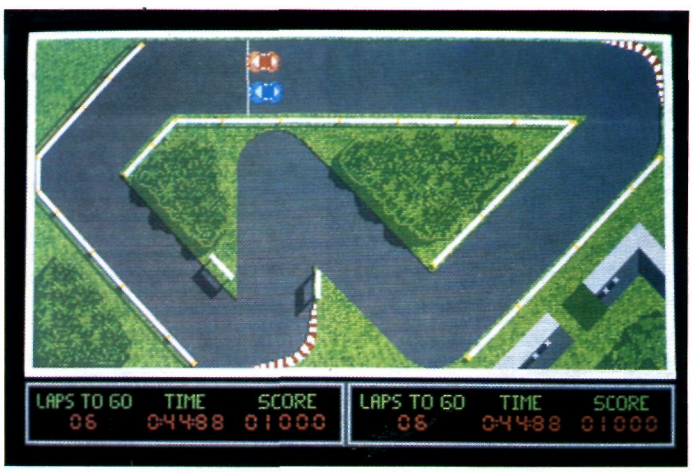
Wenn ein Kinofan den Namen 'Star Wars' hört, wird er an pakkende Science Fiction Actionsfilme erinnert, die vom amerikanischen Filmguru George Lukas produziert wurden. Die Star Wars Triologie spielte Rekordsummen in die deutschen und amerikanischen Kinokassen. Das englische Softwarehaus Domark hat die Rechte für eine »Versoftung« des Filmstoffes aufgekauft.
Vor einigen Jahren gab es schon mal ein Star Wars Game für den C64, das aufgrund der schlechten Grafik auf dem deutschen Markt keine Erfolge erzielen konnte. Ob Domark mit dem neuen Star Wars mehr Glück hat? Wir werden sehen.
In Star Wars übernehmen Sie die Rolle von Luke Skywalker. Dieser, im Film sehr sympathisch dargestellte junge Mann, hat die Aufgabe, mit einem hypermodernen Raumschiff vom Typ X-Fighter durch viele Angriffswellen zu fliegen, um schließlich zum Todesstern vorzudringen.
Dieser Todesstern bringt nichts als Unheil über die Galaxis. Ganz klar, der Todesstern muß vernichtet werden.
Im ersten Level steuern Sie Luke's Raumschiff im Anflug auf den Todesstern. Das Imperium, das den tödlichen Himmelkörper beherrscht, will es nicht zulassen, daß unser Held den Todesstern vernichtet.
Deshalb starten viele Tie-Fighter des Imperiums in den Weltraum, um Luke im Landeanflug unschädlich zu machen. Der Held sieht nur eine Möglichkeit sich zu wehren: Ballern, ballern und nochmals ballern.
Sie bewegen ein Fadenkreuz über den Bildschirm, das Sie über ein feindliches Raumschiff bewegen müssen. Haben Sie das Ziel anvisiert, drücken Sie einfach den Feuerknopf: Der böse Feind ist eleminiert.
Wenn Sie auf diese Weise genug Bösewichter abgeschossen haben, dürfen Sie im zweiten Level weiterballern. Hier haben Sie sich schon auf die Oberfläche des Planeten vorgearbeitet. Es tauchen Lasertürme aus dem Boden auf, auf die Sie permanent schießen. Auch in dieser Situation heißt es, sich wieder mit Fadenkreuz und durchtrainiertem Daumen seiner Feinde zu entledigen. Wenn Sie die Lasertürme eleminiert haben, gelangen Sie in einen Schacht, in dem der Todesstern seinen wunden Punkt hat.
Wenn Sie diesen Punkt mit einem gezielten Schuß treffen, explodiert der Todesstern und Sie haben gewonnen.
Fans von Vektorgrafik kommen an Star Wars nicht vorbei. Sie ist recht gut gelungen, schnell und fließend, wie es auf dem C64 kaum besser geht. Aber was nützt schon eine gute Vektorengrafik, wenn das Spiel so hohl ist.
Es werden keine Anforderungen an den Spieler gestellt,bis auf das Bewegen eines Fadenkreuzes über den Bildschirm und das anschließende Drückebn des Feuerknopfes. Wegen der Vektorengrafik erscheinen alle Gegner durchsichtig
Star Wars ist langweilig, vielleicht werden die angekündigten Nachfolgetitel 'The Empire strikes back' und 'Return of the Jedi' besser. cb

**Bezugsquelle**
**Elektronik von A-Z, Berlin**

| **Star Wars** | |
|---|---|
| **System: C64** | |
| **Hersteller: Domark** | |
| **Spaß/Motivation** | ★ 1 |
| **Grafik** | ★★★★★★ 6 |
| **Sound** | ★★★★★★ 6 |
| **Originalität/Idee** | ★★★★ 4 |
| **Schwierigkeit** | ★★★★ 4 |
| **RCM-Wertung** | ★★★★ 4 |

# Immer im Kreis herum

Ein Spiel wie Ralley Master von EAS Computertechnik für den Amiga ist ja schon fast eine Frechheit. Kennen Sie Supersprint von Activision? Ja? Dann kennen Sie auch Ralley Master. Ich muß Ihnen sagen, die C64 Version von Supersprint ist um Längen besser, als das Amigaspiel.
Beide haben die selbe Spielhandlung. Zwei Autos (der Spieler sieht sie auf dem Bildschirm aus der Vogelperspektive) fahren im Wettstreit einen Rennkurs, der ebenfalls nur aus der Vogelperspektive zu sehen ist. Sie können an der Leitplanke hängen bleiben und kostbare Zeit verlieren, sonst kann Ihnen nichts passieren.
Super Sprint von Activision hatte wenigstens noch einige Features, die für Abwechslung sorgten. Als einzig positiver Punkt fiel mir bei Ralley Master nur auf, daß das Programm die Highscores auf der Spieldiskette verwaltet.
Vielleicht haben Sie ja Lust, mit einem kleinen blauen Punkt, der vermutlich ein Auto darstellen soll, immer auf dem Bildschirm im Kreis herumzufahren. Ich konnte diesem Spiel nichts abgewinnen. cb




**Bezugsquelle: DTM Computerprodukte, Wiesbaden**

| **Ralley Master** | |
|---|---|
| **System: Amiga** | |
| **Hersteller: EAS** | |
| **Spaß/Motivation** | ★ 1 |
| **Grafik** | ★★ 2 |
| **Sound** | ★★★★★★★ 7 |
| **Originalität/Idee** | ★ 1 |
| **Schwierigkeit** | ★★★★★★★ 7 |
| **RCM-Wertung** | ★★★★ 4 |

# Brandheiße Spiele-Reviews

## Krieg der Sterne II

Cinemaware ist nach 'Defender of the Crown' beinahe jedem Amiga-Besitzer ein Begriff. S.D.I. ist ein weiteres Programm dieses Softwarehauses. Es ist, wie es der Name schon nahe legt, nicht mehr im Mittelalter, sondern eher der Neuzeit bzw. Zukunft angesiedelt.

Hintergrund des Spieles ist der Umsturz der sovietischen Regierung durch den KGB, der den Kurs des Politbüros für zu weich hält. Das führt dazu, daß die Welt kurz vor einem Nuklearkrieg steht. Der einzige, der das verhindern kann, sind Sie.
Natürlich gibt es auch in diesem Spiel wieder eine Frau, die der Spieler zusätzlich retten muß, bzw. soll, was in einer recht gut animierten Szene endet. Das ist allerdings nur ein Teil der Aufgabe. Auf dem Weg dahin hat man zahlreiche Hindernisse zu überwinden.
Da wäre zum einen die Order, sämtliche von der UDSSR abgefeuerten Raketen mittels Satelliten zu zerstören. Diese Satelliten maß man außerdem ständig funktionsfähig halten, da russische Raumschiffe diese dauernd beschießen und so beschädigen.
Anfallende Reparaturen sowie das Eindringen in die russische Raumstation bewältigt man mittels eines Raumschiffs, wobei der Flugsimulator recht gut und schnell programmiert ist. Der Einflug in die eigene, bzw. fremde Station wurde mittels 3D-Drahtgitter Animation erreicht, wobei sehr fließende Animationen erscheinen.
Auffällig bei diesem Spiel sind wieder, wie bei Cinemaware nicht anders zu erwarten, die sehr gut gelungenen Zeichnungen (die müssen echt den ganzen Tag nur malen...). Auch wurden die Animationen der Figuren sehr gut programmiert. Desweiteren gefiel uns die Tatsache, daß man vor dem Spielen gefragt wird, ob eine Ram-Disk angelegt werden soll, was bei positiver Antwort und ca. 2 Minuten Ladezeit dazu führt, daß alle anschließenden Ladeoperationen nicht mehr von Disk sondern von Ram erfolgen, was den Spielzusammenhang nicht so stark zerreißt.
Auch finde ich es gut, daß eine Quick Reference Card mitgeliefert wurde, wodurch man einzelne Teile noch einmal nachlesen kann, ohne lange im Manual zu suchen.
Abschließend kann man sagen, daß es sich grafisch um ein sehr gutes Spiel handelt, daß trotz seines Alters und des recht kriegerischen Hintergrunds den Spieler gut motivieren kann. gg

**Bezugsquelle:**
**Intersoft LaSch, Oberhausen**

**S.D.I.**
**System: Amiga**
**Hersteller: Cinemaware**

| | |
|---|---|
| **Spaß/Motivation** | ★★★★★★ **6** |
| **Grafik** | ★★★★★★★ **7** |
| **Sound** | ★★★★★★★ **7** |
| **Originalität/Idee** | ★★★★★★ **6** |
| **Schwierigkeit** | ★★★★★★ **6** |
| **RCM-Wertung** | ★★★★★★ **6** |

## Männer und Motoren

Autorennen gibt es eine ganze Menge für den Amiga. Ich denke da beispielsweise an Crazy Cars oder Test Drive. Gute Grafik haben Sie auch alle. Aber Grid Start ist bisher das erste Autorennen, das die spannende Atmosphäre vermitteln kann.
Professional, Amateur und Novice sind die drei Klassen, die in der Highscoreliste berücksichtigt werden. Zuerst würde ich Ihnen empfehlen, Novice zu wählen. Nach der Wahl des Schwierigkeitsgrades können Sie sich noch einen von sechs Kursen auswählen, auf dem das Rennen stattfinden soll. Es stehen Brands Hatch, Silver Stone, Belgium, Mexico, Austria und Germany zur Auswahl.
Die Startampel springt von Rot auf Grün. Die Reifen quietschen. Die rasante Fahrt geht los. Sie befinden sich mit Ihrem Wagen an Position 24 und müssen alle anderen Wagen einholen. Ihr Rennwagen verfügt über sechs Gänge, zwischen denen Sie immer hin- und herschalten müssen.
Wenn Sie einen Wagen rammen oder aus der Kurve fliegen, gibt es eine grafisch sehr gut dargestellte Explosion.
Überhaupt ist die Grafik insgesamt sehr gut. Die Sounds, also das digitalisierte Reifengequietsche und die Motorgeräusche wissen ebenfalls voll zu überzeugen, weshalb wir uns bei unserer Bewertung auch auf die Höchstnote 7 geeinigt haben. Mag sein, daß wir im Laufe der Monate kritischer werden müssen. Noch überwiegt jedoch der Stauneffekt. cb

**Bezugsquellen:**
**Soyka Datentechnik, Bochum**
**Elektronik von A-Z, Berlin**

**Grid Start**
**System: Amiga**
**Hersteller: Anco**

| | |
|---|---|
| **Spaß/Motivation** | ★★★★★★★ **7** |
| **Grafik** | ★★★★★★★ **7** |
| **Sound** | ★★★★★★★ **7** |
| **Originalität/Idee** | ★★★★★★ **6** |
| **Schwierigkeit** | ★★★★★★ **6** |
| **RCM-Wertung** | ★★★★★★★ **7** |

was will man mehr...

Rainbow Arts

## JINKS

Mit einem dreieckigen Raumgleiter muß die runde Sonde über die scrollenden Landschaften von Atavi geschossen werden. Die Flugbahn der Sonde wird sich jedoch schneller ändern, als Ihnen lieb sein kann. An vielerlei Hindernissen kann diese abprallen, starke Magnetfelder können sie ablenken, in den kochenden Sümpfen kann die Sonde ein klägliches Ende finden. Dann gilt es schnell zu handeln, die Sonde muß wieder auf die richtige Bahn gebracht werden.

Erhältlich für:
COMMODORE 64, AMIGA

## STREET GANG

Mickys Leben war bisher ebenso normal wie auch langweilig. Mit seinen Eltern wohnte er in einer friedlichen Kleinstadt auf dem Land. Doch dann kam der Tag, an dem sich alles ändern sollte. Wild entschlossen zogen seine Eltern nach New York, dort werden die Straßen von den verschiedensten Banden regiert. Da Micky plötzlich seine coole Ader entdeckte, wollte er unbedingt Mitglied in einer dieser Banden werden. Doch um Aufnahme zu finden, muß er eine schwierige Mutprobe bestehen. Und so macht Micky sich auf den Weg, der berüchtigten Locke sein heißgeliebtes Haarbüschel zu rauben.

Erhältlich für:
C 64, ATARI ST – XE/XL, AMSTRAD, AMIGA

## ANTICS

Der normalen Programmierarbeit überdrüssig kramte das ARTGAMES Team mal wieder in der Archivkiste: Dort entdeckte es vier ältere, bisher völlig unveröffentlichte Spiele. Diese Spiele sind klein, kompakt und machen sehr viel Spaß, aus diesem Grunde wurde der Entschluß gefaßt, durch Neuprogrammierung, neue Grafik und fantastischen Sound die Programme auf den heutigen Stand zu bringen. ANTICS ist also ein Sampler, der sich durch Originalität und Spielwitz auszeichnet.

Erhältlich für:
COMMODORE 64

## IN 80 DAYS AROUND THE WORLD

Die Faszination einer Weltreise, der Reiz fremder Kulturen, die Erforschung unheimlicher Plätze machen dieses Spiel zu einem fesselnden Erlebnis. Die Handlung orientiert sich am gleichnamigen Weltbestseller von JULES VERNE. In einem edlen englischen Club schließt der leicht angetrunkene Philias Fogg eine unglaubliche Wette ab. Er behauptet, die Erde im Jahre 1872 ohne fremde Hilfe zu umrunden. Diese Wette, von den Clubmitgliedern höhnisch belächelt, entwickelt sich zu einem Abenteuer auf Leben und Tod.

Erhältlich für:
COMMODORE 64, ATARI ST – XE/XL, AMIGA

## BAD CAT

Los Angeles, 1984. Die Stadt rüstet sich für die olympischen Sommerspiele. Große Leute halten ebenso große Reden, alles ist gerichtet für die Besuchermassen. Doch da sind noch die streunenden Großstadtkatzen, bisher von allen mißachtet, organisieren sie eine Konkurrenzveranstaltung. Keiner wird dieses Gebaren der Katzen so schnell vergessen, denn überall in der Stadt treffen sich wilde Katzenhorden um die ausgefallensten Wettkämpfe zu veranstalten. Wird BAD CAT ihrem Namen alle Ehre machen?

Erhältlich für:
COMMODORE 64, AMIGA

## THE GREAT GIANA SISTERS

Die kleine Giana schläft tief und fest. Plötzlich beginnt sie zu träumen, sie befindet sich in einer völlig verrückten Fantasiewelt. Unheimliche Wesen begegnen Ihr dort, die Landschaften sind voller Tücken und Geheimnisse. Dauernd entdeckt Sie neue Geheimgänge, wundersame Pflanzen und Kreaturen. Doch das eigentlich Schlimme ist die Tatsache, daß Sie aus dem Traum nur wieder zurückkehren kann, wenn Sie die mysteriöse Landschaft komplett durchquert. Da sollte ein Gentleman doch wohl helfen – oder?

Erhältlich für:
COMMODORE 64, AMIGA, ATARI ST

# INFO-Coupon

Gegen Einsendung dieses Coupons und DM 5,– in Briefmarken erhalten Sie unseren Gesamtkatalog sowie ausführliches Infomaterial.

Bitte Computertyp ankreuzen:

AMIGA ☐
ATARI-ST ☐
COMMODORE 64 ☐
SCHNEIDER CPC ☐

Name: _____

Straße/Nr.: _____

PLZ/Ort: _____

Bitte sofort absenden an:

RUSHWARE GMBH
BRUCHWEG 128 - 132
4044 KAARST 2

# Umsteigen leicht gemacht

## Ein Kurs für Ein- und Aufsteiger (Teil 3)

**Im letzten Teil haben wir uns die Geschwindigkeit von C vor Augen geführt. Doch warum sollen wir diese nur für eher witzlose Zeittests oder CLI-Befehle benutzen? Ab heute werden wir die tolle Grafik des Amiga ausnutzen:**

Der Amiga hat durch sein ausgeklügeltes Grafiksystem aus Windows und Screens Möglichkeiten, die ein anderer Computer erst durch komplizierte Berechnungen erreicht.
Allerdings werden vom Benutzer eine Reihe Vorarbeiten und Routinen verlangt, und die sorgen dafür, daß jedes C-Programm von vornherein einige KBytes lang ist. Aber die Möglichkeiten, die man dann hat, sind schlicht überragend: beliebige Zeichensätze, Flächen und Linien mit Mustern, hardwaremäßiger Floodfill, automatisches Clipping, usw. - kurz gesagt, alles was man für Supergrafiken braucht.
Um einmal am »eigenen Leib« zu erfahren, wie schnell die Grafik in C ist, haben wir im Listing unten ein Programm abgedruckt, das ein Grundstein für einfache Grafikprogramme darstellt. An der gekennzeicheneten Stelle im Listing können eigene Routinen eingesetzt werden. Dafür hier die wichtigsten Grafik-Befehle:

**Move (rp,x,y)**

Dieser Befehl bewegt den Zeichenstift (Pixelcursor) zur angegeben Stelle, ohne zu zeichnen.

**Draw (rp,x,y)**

Hiermit zeichnet man eine Linie von der aktuellen Cursorposition zum Punkt x,y.

**SetAPen (rp,c)**

Ändert die aktuelle Zeichenfarbe in Farbe c um.

**SetDrMd (rp,mode)**

Mit SetDrMd ändert man den aktuellen Zeichenmodus. Dies sind:

| | |
|---|---|
| **JAM1:** | nur mit Vordergrundfarbe malen |
| **JAM2:** | mit Vordergrundfarbe malen, Rest mit Hintergrundfarbe ausmalen |
| **COMPLEMENT:** | komplementiert Zeichen mit dem Hintergrund |
| **INVERSEVID:** | invertiert die Grafik. Vorder- und Hintergrundfarbe werden vertauscht. |

**Text (rp,"(Text)",TextLen)**

Wie der Befehl schon sagt, kann man hiermit Text ausdrucken. TextLen ist hierbei die Länge des auszudruckenden Textes.

Nun ein Hinweis zur Benutzung: Man sollte sich bei Benutzung von Aztec C, gleich von Anfang an daran gewöhnen, alle Betriebssystemaufrufe mit LONG-Parametern zu machen, z.B.:
**Move (rp,100L,20L)**
oder:
**Draw (rp,(long)x,(long)y)**

**rp** bezieht sich bei allen Befehlen auf den RastPort, in dem gezeichenet werden soll. (Im Grundprogramm wird rp schon definiert).

An dieser Stelle sollten Sie nun erst einmal das Grundprogramm (Listing am Kursteilende) abtippen. Legen Sie es am besten unter einem Namen wie zum Beispiel 'gg.c' (für Graphik Grundprogramm) auf der Programmdiskette ab. Sicherheitshalber sollten Sie es mittels'protect gg.c re' vor Löschen und Überschreiben schützen.
Danach probieren Sie bitte noch Listing 2 aus, das Sie ab der Zeile 41 main () (anstatt 41-62) im Grundprogramm einfügen müssen.
Nach dem Compilieren können Sie das Programm starten und die Geschwindigkeit bewundern. Beenden kann man das Programm mittels Anklicken des unsichbaren Close-Gadgets in der linken oberen Ecke.
Probieren Sie nun, folgende Aufgabe zu lösen: Zeichnen Sie Linien von der linken Seite des Bildschirms durch den Mittelpunkt zur rechten Seite; danach von der oberen Seite durch den Mittelpunkt zur unteren, diesmal mit Invertieren der Linien. Und zum guten Schluß noch Linien um den Mittelpunkt mit einem Radius von 75 in der Farbe 2 mit invertieren. Auf die Plätze, fertig, los ...

Na, geschafft? Wenn ja, dann herzlichen Glückwunsch. Falls nicht, dann trösten Sie sich damit, daß wir alle mal diese und ähnliche Probleme hatten und sie ebenfalls nicht lösen konnten. Probieren es noch einmal, es klappt bestimmt.

## Der Aufbau der Graphik

Aufsteiger vom C64 fragen sich, wieso kann der Amiga so viele Farben darstellen.
Beim C64 ist die Multicolor-Grafik so aufgebaut, daß jeweils 2 nebeneinanderliegende Pixels innerhalb eines 8x8-Pixelblocks die Farbe bestimmen. Wenn man dies auf den Amiga übertragen würde, so würde das bedeuten, daß zum Darstellen der 'normalen' maximalen Farbanzahl von 64 Farben im ExtraHalfBrite-Modus 6 Bits nebeneinander benötigt würden. Das ist schon vom Errechnen einer bestimmten Position auf dieser benötigten Riesen-Bitmap viel zu umständlich.
Also wird es wohl anders gehen: Die Lösung; die Bits werden nicht nebeneinander sondern hintereinander angeordnet, wodurch die einmal gewählte volle Auflösung erhalten bleiben kann.
Dies wird in Bild 1 gezeigt. Hier sieht man, wie die einzelnen Bitplanes hintereinander liegen, womit man die einzelnen verschiedenen unabhängigen Farben darstellen kann.
Aber der Amiga kann doch bis zu 4096 Farben auf einmal darstellen! Genau darauf bezog sich im letzten Satz das Wort unabhängig, denn der Amiga kann maximal 32 Farben darstellen, die voneinander total unabhängig sind. In allen anderen Modi wird eine größere Farbanzahl nur durch Tricks erreicht, indem sich das System an die 32 Grundfarben anlehnt.

## Der Halfbrite Modus (EHB)

Der Amiga hat in seinem Grafikchip genau 32 Farbregister, d.h. daß man eigentlich auch nur 32 Farben darstellen kann. Wie geht das aber im EHB, wo doch 64 Farben möglich sind?
Wie das 'Halfbrite' (auf deutsch in etwa 'halb so hell') schon sagt, werden hier die 'normalen' 32 Farben noch einmal mit halber Intensität dargestellt, so daß man bei geschickter Farbwahl wirklich 64 verschiedene Farben hat. An dieser Stelle daher ein Aufruf an die Programmierer:Schicken Sie uns doch ein Programm, daß ein HAM-Bild in den EHB-Modus umrechnet!

## Der Hold-And-Modify Modus (HAM)

Auch in diesem Modus sind die Farben nicht unabhängig, denn man bräuchte für solch eine Farbschwemme 12 Bitplanes, dann allerdings keine Farbregister mehr, da jede Bitkombination der jeweiligen Farbe entspricht.
Der Amiga benötigt allerdings nur 6 Bitplanes, um diese Farben darzustellen, wie geht das dann?
Im HAM haben die 6 Bitplanes verschiedene Bedeutungen: Die oberen 2 Bits haben eine Steuereigenschaft, die unteren 4 dann davon abhängig eine bestimmte Bedeutung.
Sind die obersten 2 Bit '0', so wird die Farbe für den Punkt direkt aus dem Farbregister, dessen Nummer in den untersten 4 Bit steht, genommen, d.h. daß 16 Farben direkt und ohne 'Tricks' darstellbar sind. Wie kommt man nun zu den übrigen 4080 Farben? Wenn die obersten 2 Bit nun einen Wert von '10','01' oder '11' enthalten, haben die unteren 4 Bit einen vollkommen andere Bedeutung: Je nach den beiden Steuerbits wird entweder der Rot-, Grün- oder Blau-Anteil des vorhergehenden Bits auf den Wert der unteren 4 Bits gesetzt, die anderen beiden Anteile bleiben erhalten.Um z.B. von Schwarz zu Weiß zu kommen, sind also 3 Schritte nötig:

| | | | | |
|---|---|---|---|---|
| R: | 0 | F | F | F |
| G: | 0 | 0 | F | F |
| B: | 0 | 0 | 0 | F |

Das resultiert dann in den typischen Streifen links von harten Farbübergängen.
Damit ist unser heutiger Teil des Kurses beendet. Wir wünschen Ihnen viel Spaß beim Ausprobieren der tollen Grafik.
Weitere Teile unseres Kurses werden in loser Folge erscheinen. So planen wir einen Teil über Maschinenprogrammierung, Interruptbehandlung und Devices.

---

Achtung!
Wer im letzten Heft das Echo-Programm abgetippt hat, wird festgestellt haben, daß irgend etwas nicht gestimmt haben kann. Richtig, es fehlte das Ende des Listings. Die Korrektur dazu befindet sich in diesem Heft an anderer Stelle.

Wir bitten diesen Fehler zu entschuldigen.

```
/**********************
 Graphics Grundprogramm
      (C) 1987 by
 Roeske's Computer Mag.
 **********************/

#include <exec/types.h>
#include <graphics/gfxbase.h>
#include <intuition/intuition.h>

#define XSIZE          320
#define YSIZE          256
#define ever           (;;)

struct IntuitionBase   *IntuitionBase;
struct GfxBase         *GfxBase;
struct Screen  *scr, *OpenScreen();
struct Window  *win, *OpenWindow();
long GetMsg();


struct NewScreen scrdef = {
   0, 0, XSIZE+1, YSIZE+1,                 /* Größe der Screen */
   4,                                      /* Bitplanes */
   0, 1,
   ((XSIZE>320) ? HIRES : 0) | ((YSIZE>256) ? LACE : 0),
   CUSTOMSCREEN,                           /* eigene Screen */
   NULL, NULL, NULL, NULL };

struct NewWindow windef = {
   0, 0, XSIZE+1, YSIZE+1,                 /* Größe des Windows */
   0, 1,
   CLOSEWINDOW,                            /* IDCMP Flags */
   WINDOWCLOSE | BACKDROP | BORDERLESS,    /* Flags */
   NULL, NULL,
   NULL,                                   /* Titel */
   NULL,                                   /* Screen */
   NULL,0, 0, 0, 0,
   CUSTOMSCREEN };                         /* eigene Screen */

main ()
{
   struct ColorMap  *color_map;
   register struct RastPort *rp;
   register struct ViewPort *vp;
   openstuff();
   vp = & (scr -> ViewPort);
   rp = & (scr -> RastPort);
   SetDrMd (rp, JAM1);
   ShowTitle (scr, FALSE);
/*              Punkt: WritePixel(rp,(long)x,(long)y);
   52             Linie:   Move(rp,sx,sy);Draw(rp,ex,ey);
   53             Farben:  SetAPen(rp,farbe);
   54                      SetBPen(rp,farbe);
   55             BSloesch: SetRast(rp,farbe);
   56             Modus:   SetDrMd(rp,JAM1/JAM2);
   57             Text:    Move;Text(rp,"(text)",len(text)) */
   WaitPort(win->UserPort);
   ex();
   /* In einer Schleife lautet die Abfrage des Gadgets so:
   if (GetMsg (win -> UserPort)) ex();
 */
}

openstuff ()
```

**Listing 1**

```
65  {
66     char *OpenLibrary();
67
68     if (!(IntuitionBase = (struct IntuitionBase *)
69        OpenLibrary ("intuition.library", 1L))) {
70           printf ("keine Intuition-Library\n");
71           ex();
72     }
73     if (!(GfxBase = (struct GfxBase *)
74        OpenLibrary ("graphics.library", 1L))) {
75           printf ("keine Graphics-Library\n");
76           ex();
77     }
78     if (!(scr = OpenScreen (&scrdef))) {
79           printf ("Kein Speicher für Screen\n");
80           ex();
81     }
82     windef.Screen = scr;
83     if (!(win = OpenWindow (&windef))) {
84           printf ("Kein Speicher für Window\n");
85           ex ();
86     }
87  }
88
89  ex ()
90  {
91  if (win)    CloseWindow (win);
92  if (scr)    CloseScreen (scr);
93  if (GfxBase)    CloseLibrary (GfxBase);
94  if (IntuitionBase)    CloseLibrary (IntuitionBase);
95  }
96
```

**Fortsetzung von Listing 1**

```
41  main ()
42  {
43     struct ColorMap  *color_map;
44     register struct RastPort *rp;
45     register struct ViewPort *vp;
46     long t;
47     openstuff();
48     vp = & (scr -> ViewPort);
49     rp = & (scr -> RastPort);
50     SetDrMd (rp, JAM1);
51     ShowTitle (scr, FALSE);
52     SetRast (rp,0L);            /* RastPort löschen *
/
53     SetAPen(rp,3L);
54     for (t=0;t<YSIZE;t+=2) {
55              Move(rp,0L,0L);
56              Draw(rp,(long)XSIZE,(long)(YSIZE-t));
57     }
58     SetAPen(rp,1L);
59     for (t=0;t<XSIZE;t+=2) {
60              Move(rp,t,0L);
61              Draw(rp,t,(long)YSIZE);
62     }
63     SetAPen(rp,2L);
64     for (t=0;t<YSIZE;t+=2) {
65              Move(rp,0L,t);
66              Draw(rp,(long)XSIZE,t);
67     }
68     WaitPort(win->UserPort);     /* auf Message warten
 */
69     ex();                                  /* Raus */
70  }
```

**Listing 2**

# C-Kurs für Einsteiger Teil 2

Nachdem wir uns im letzten Kurs hauptsächlich mit dem Aufbau eines C-Programmes und den Variablenarten begnügt haben, werden wir uns in diesem Teil, mit der Ausgabe auf den Bildschirm (printf) und mit der Eingabe über die Tastatur (scanf) unter C beschäftigen.
Direkte Stringausgabe mit printf

In einem Programm kommt es vor, dass man eine Meldung zum Monitor ausgeben möchte: Beispielsweise die Meldung: 'Drücken Sie die rechte Maustaste!' So eine Meldung lässt sich einfach mit printf realisieren, wie das folgende Beispielprogramm 1 (bitte wie alle Amiga-Programme ohne Zeilennummern eintasten) zeigen wird:

```
/* Programm zur direkten Stringausgabe */
main()
{
printf("Druecken Sie die Maustaste");
}
```

Wie Sie bereits aus Kursteil 1 wissen, handelt es sich bei der 1. Zeile unseres Beispielprogrammes um eine Kommentarzeile. Zeile 2 benennt die Hauptfunktion (main), die in der 3. Zeile durch die geschweifte Klammer eröffnet wird. In der Zeile 4 schließlich wird die Meldung mit Hilfe von printf ausgegeben. Wie Sie aus dem Programm ersehen können, steht hinter dem Ausgabestring noch ein sogenanntes Backslash und ein n . Man nennt dies eine Fluchtsymbolfolge (engl.: 'escape sequence'); der Backlash nennt sich entsprechend -Fluchtsymbol-.

Fluchtsymbolfolgen:

```
\b - Ruecktaste
\f - Seitenvorschub
\n - Zeilentrenner (z. Anf. d. naechsten Zeile)
\r - Wagenruecklauf
\t - Tabulator
\\ - Backslash
\0 - Null
\' - einfaches Anfuehrungszeichen
\" - doppeltes Anfuehrungszeichen
```

Fluchtsymbolfolgen können jederzeit innerhalb des Ausgabestrings vorkommen. Doch nun zur Ausgabe vor variablen Daten.

Variablenausgabe mit printf

In unserem Computer sind sämtliche Variable binaer im Speicher abgelegt. Verschiedene Datentypen werden allerdings auf unterschiedliche Art und Weise gespeichert. Üblicherweise werden Zahlen, egal in welcher Art sie vorliegen (int, float...) dezimal ausgegeben. Dazu muss man aber der auszugebenden Routine sagen, um welchen Datentyp es sich handelt, damit eine entsprechende Umwandlung stattfinden kann.

-printf- braucht um Variablen ausgeben zu können, also zwei Parameter:

1.) Wie soll es ausgegeben werden?
2.) Welche Variable(n) soll(en) ausgegeben werden?

zu 1:

Dies wird mit einem sogenannten Kontrollstring festgelegt. Wie der Name schon andeutet, handelt es sich hierbei um eine Zeichenkette, die von je zwei Anführungszeichen umschlossen ist. Sie beinhaltet ein oder mehrere Umwandlungsvorschriften und eventuell vorhandene Fluchtsymbol- und Textfolgen.

Eine Umwandlungsschrift beginnt stets mit dem Prozentzeichen (%). Diesem kann eine Zahl folgen und daran schliesst sich ein Umwandlungszeichen (siehe Tabelle 2.2) an. Zu jeder auszugebenden Variable muss eine Umwandlungsvorschrift vorhanden sein. Sie muss in der Reihenfolge auftauchen, wie die Variablen, die ausgegeben werden.

Die Bedeutung der Umwandlungsvorschrift:
Eine Umwandlungsvorschrift wird in einem Kontrollstring durch das erste Zeichen, das %-Zeichen erkannt. Durch die darauffolgende Zahl wird die Länge eines Datenfeldes bestimmt, in der die Variable ausgegeben werden soll. Innerhalb dieses Feldes wird die Zahl rechtsbündig ausgegeben.

Bsp.: Eine vierstellige Zahl soll in einem zehnstelligen Feld ausgegeben werden:
Ausgegeben wird:
Sechs Leerzeichen, gefolgt von der vierstelligen Zahl, die angezeigt werden soll.

Wie oben schon angedeutet, ist die Längenangabe optional. In diesem Fall wird die Feldbreite automatisch so gewählt, dass die ganze Zahl ausgegeben werden kann.

Der Buchstabe, der sich an die Zahl anschließt, ist ein Umwandlungszeichen. Er gibt an, wie die als Binaerzeichen gespeicherte Information ausgegeben werden soll.

Umwandlungszeichen:

```
b   - Binaerzahl
c   - einzelnes Zeichen
d   - dezimale Integer
e   - Gleitkommazahl mit Exponent
f   - Gleitkommazahl ohne Exponent
g   - entweder e oder f, das kuerzere
h   - Integer
ld  - long Integer
o   - Oktalzahl
s   - Zeichenkette
u   - vorzeichenlose Integer
x   - Hexadezimalzahl
```

Für Gleitkommazahlen sind drei Umwandlungszeichen vorhanden. Wird ein e benutzt, so wird die Zahl mit einem Exponenten ausgegeben. Ein f bewirkt eine normale Ausgabe ohne Exponent. Gebrauchen wir als Umwandlungszeichen ein g, so wirkt dies, als hätten wir e oder f gewählt. Die Entscheidung über die Art der Ausgabe erfolgt während des Programmlaufes. Es wird dann immer die kürzere der beiden Möglichkeiten verwendet.

Beispielprogramm 2

```
/* Programm zum Ausgeben zweier floats */
main()
        float a, b;
        a = 12.3;
        b = 123000000;
        printf("a= %10e\n",a);
        printf("a= %10f\n",a);
        printf("a= %10g\n",a);
        printf("b= %10e\n",b);
        printf("b= %10f\n",b);
        printf("b= %10g\n",b);
}
```

Man kann dieses Programm natürlich auch kürzer schreiben:

Beispielprogramm 3

```
/* Programm 2 zum Ausgeben zweier floats */
main()
        float a, b;
        a = 12.3
        b = 123000000
        printf("a=%10e\na=%10f\na=%10g\nb=%10e\nb=%10f\nb=
%10g\n",a,a,a,b,b,b);
}
```

Wie Sie selbst sehen können, verliert diese Schreibart an Übersichtlichkeit und sollte deshalb vermieden werden. Natürlich können auch mehrere Fluchtsymbolfolgen hintereinander auftreten. Auch dürfen verschiedenartige Variablen gemischt werden. Es muss jedoch immer darauf geachtet werden, daß die Anzahl und die Reihenfolge der Umwandlungszeichen und der Variablen übereinstimmt.

scanf - eine Eingabefunktion

Die scanf-Anweisung ähnelt stark der printf-Anweisung, bis auf einen sehr wichtigen Unterschied: Das printf-Kommando enthält eine Reihe von Variablennamen. Bei der Benutzung von scanf hingegen, müssen die Anfangsadressen der Variablen angegeben werden. Glücklicherweise bietet C dem Programmierer eine einfache Möglichkeit, diese zu erhalten:
Wird in C vor den Variablennamen ein kaufmännisches Und (&) vorgesetzt, so erhält man auf diese Weise die Adresse der angegebenen Variablen. Man nennt dies auch Zeiger der Variablen, oder einfach Zeiger (engl.: pointer). Der Lattice C-Compiler erlaubt, diese Variablen als 'pointer' zu definieren. Beim weniger komfortablen C-Compiler müssen diese Variablen als 'unsigned int' definiert werden. Der dann gespeicherte Wert entspricht der ersten Speicherstelle, bei der die Speicherung der Variablen beginnt.
Beispielprogramm 4

```
/* Programm mit scanf-Eingabe */
main()
        {
                char a;
                scanf("\nSie sind %2c Jahre alt.\n",a);
        }
```

Sicher ist es auch möglich, gleichzeitig mehrere Variablen auf einmal einzulesen. Und dies geht auf folgendem Weg:
Beispielprogramm 5

```
/* Programm 2 mit scanf-Eingabe */
/* Ellipsenberechnung */
#define PI 3.1415926536
main()
{
        float r, x, y, f;
        printf("\nGeben Sie den x- und y-Radius in cm an!"
);
        scanf("\n%f%f",&x,&y);
        r = (x + y)/2;
        f = r * r * PI;
        printf("\nDie Flaeche = %g qcm.\n",f);
}
```

Zeile 3 beinhaltet eine neue Anweisung ausserhalb der main-Funktion. Sie ist eine Anweisung an den Preprozessor von C. Und das geht folgendermassen:
Bevor das Programm vom Compiler übersetzt wird, werden alle mit 'define' erzeugten Konstanten im Programm durch ihren definierten Wert ersetzt.
Das bedeutet, daß die Zeile 8 während des Compilerlaufes so aussieht:
8 f = r * r * 3.1415926536
Zu den Zeilen 6 und 9 ist noch zu sagen, daß sich die Länge des einzugebenden Feldes automatisch definiert. Jedoch muß zwischen beiden Eingaben ein Trennzeichen sein. In C kann dies sein:

1. - ein Leerzeichen
2. - ein Tabulator
3. - ein Return

Doch kurzer Rede, langer Sinn. Am besten Sie probieren es selbst einmal aus. Mit 'printf' und 'scanf' können Sie schon jede Menge Programme schreiben. Tun Sie das, denn nur die Praxis macht den Meister.

In unserem nächsten Heft werden wir uns dann mit Schleifen und Wiederholmöglichkeiten beschäftigen, so daß schon bald umfangreichere und auch wirklich komplexere Programme entwickelt werden können. (HF)

# Critical Mass

Das Spiel ist in Amiga-BASIC geschrieben und läuft auf jedem Amiga mit mindestens 512 KB und benötigt keinerlei zusätzliches Programme oder Libraries.Es kann mit dem Amiga-BASIC Editor abgetippt werden und läßt sich von der WorkBench aus laden.
Das Programm ist ein Strategiespiel für zwei Personen. Diese setzen abwechselnd einen roten bzw. blauen Stein auf ein eigenes oder leeres Feld. Befinden sich auf einem Feld so viele Steine wie es Nachbarfelder hat, so explodiert es.
Wenn ein Feld explodiert, schleudert es alle Steine auf die Nachbarfelder. Diese und die gegnerischen Steine, die sich darauf befinden, gehören nun zu den eigenen. So können ganze Kettenreaktionen entstehen.
Sind nur noch Steine eines Spielers auf dem Spielfeld vorhanden, so hat dieser gewonnen.

```
   1  SCREEN 2,320,200,2,1
   2  WINDOW 2,"Critical Mass",,16,2
   3  PALETTE 0,0,.61,.31
   4  PALETTE 1,0,0,0
   5  PALETTE 2,1,0,.33
   6  PALETTE 3,0,1,1
   7  GOSUB Start
   8  CLS:RANDOMIZE TIMER:DEFINT a-z:s=1:z=1:sp=1
   9  DIM f(500),g(500)
  10  FOR i=0 TO 160 STEP 16
  11    LINE (0,i)-(160,i),1
  12    LINE (i,0)-(i,160),1
  13  NEXT i
  14  Namen:
  15  p=3:t=1
  16  FOR i=1 TO 2
  17    Sprung0:
  18    GOSUB Loeschen
  19    COLOR 1,0
  20    LOCATE 5,22:PRINT "Name von Spieler";i
  21    COLOR p,0
  22    LOCATE 7,22:INPUT a$(t)
  23    IF LEN(a$(t))<1 THEN a$(t)="Spieler"+STR$(i)
  24    IF LEN(a$(t))>15 THEN a$(t)=LEFT$(a$(t),15)
  25    p=2:t=3
  26  NEXT i
  27  IF RND<.5 THEN a=1:f1=2 ELSE a=-1:f1=3
  28  FOR b=1 TO 10
  29    FOR c=1 TO 10
  30      d(b,c)=0
  31    NEXT c
  32  NEXT b
  33  Sprung4:
  34  c=0:b=0:x=0:e=0:a=a*(-1):f1=f1-a
  35  GOSUB Loeschen
  36  COLOR 1,0
  37  LOCATE 5,22:PRINT "An der Reihe ist:"
  38  IF a=1 THEN COLOR 2,0 ELSE COLOR 3,0
  39  LOCATE 7,22:PRINT a$(a+2)
  40  Abfrage:
  41  IF sp=-1 THEN Maus
  42  Joystick:
  43    m=STRIG(2):FOR vz=1 TO 250:NEXT vz
  44    IF STICK(2)=1 THEN s=s+16 ELSE IF STICK(2)=-1 THEN
s=s-16
  45    IF STICK(3)=1 THEN z=z+16 ELSE IF STICK(3)=-1 THEN
z=z-16
  46    IF s<1 THEN s=1 ELSE IF s>159 THEN s=159
  47    IF z<1 THEN z=1 ELSE IF z>159 THEN z=159
  48    GOTO Feld
  49  Maus:
  50    m=MOUSE(0):s=MOUSE(1)-1:z=MOUSE(2)
  51    IF s<1 OR s>159 OR z<1 OR z>159 THEN Maus
  52    IF s/16=INT(s/16) THEN s=(c2-1)*16:m=0
  53    IF z/16=INT(z/16) THEN z=(b2-1)*16:m=0
  54  Feld:
  55  c=INT(s/16)+1:b=INT(z/16)+1:mx=(c-1)*16:my=(b-1)*16
  56  IF c2<>c OR b2<>b THEN LINE (mx2,my2)-(mx2+16,my2+16)
,1,b
  57  LINE (mx,my)-(mx+16,my+16),f1,b
  58  c2=c:b2=b:mx2=mx:my2=my
  59  IF m<>0 THEN GOTO Setzen
  60  GOTO Abfrage
  61  Setzen:
  62  SOUND 493.88,5
  63  IF a*ABS(d(b,c))=d(b,c) THEN d(b,c)=d(b,c)+a ELSE Abf
rage
  64  sp=sp*(-1)
  65  GOSUB Sprung1:GOSUB Sprung2
  66  IF e=1 THEN Sprung5
  67  Warten:
  68  IF MOUSE(0)<>0 THEN Warten ELSE Sprung4
  69  Sprung5:
  70  x=x+1:b=f(x):c=g(x):d(b,c)=0
  71  GOSUB Sprung1
  72  b=b-1:IF b>0 THEN GOSUB Sprung3
  73  b=b+2:IF b<11 THEN GOSUB Sprung3
  74  b=b-1:c=c-1:IF c>0 THEN GOSUB Sprung3
  75  c=c+2:IF c<11 THEN GOSUB Sprung3
  76  IF x/10=INT(x/10) THEN Sprung6 ELSE IF x<e THEN Sprun
g5
  77  Sprung6:
  78  k=0:j=0
  79  FOR c=1 TO 10
  80    FOR b=1 TO 10
  81      k=k+d(b,c):j=j+ABS(d(b,c))
  82    NEXT b
  83  NEXT c
  84  IF ABS(k)<>j AND x=e THEN Warten ELSE IF ABS(k)<>j TH
EN Sprung5
  85  GOSUB Loeschen
  86  COLOR 1,0
  87  LOCATE 5,22:PRINT "Gewonnen hat:"
  88  COLOR f2,0
  89  LOCATE 7,22:PRINT a$(a+2)
  90  COLOR 1,0
  91  LOCATE 22,1:PRINT "Noch ein Spiel (J/N)"
  92  Taste1:
  93  t$=INKEY$:IF t$="n" THEN Ende ELSE IF t$="j" THEN RUN
 ELSE Taste1
  94  Sprung2:
```

```
   95  IF c=1 OR c=10 THEN Sprung8 ELSE IF b=1 OR b=10 THEN
Sprung9 ELSE IF ABS(d(b,c))=4 THEN Sprung7 ELSE RETURN
   96  Sprung8:
   97  IF b=1 OR b=10 THEN IF ABS(d(b,c))=2 THEN Sprung7 ELS
E RETURN
   98  Sprung9:
   99  IF ABS(d(b,c))=3 THEN Sprung7 ELSE RETURN
  100  Sprung7:
  101  e=e+1:IF e=200 THEN Sprung6 ELSE f(e)=b:g(e)=c:RETURN
  102  Sprung3:
  103  d(b,c)=a*ABS(d(b,c))+a
  104  GOSUB Sprung2
  105  Sprung1:
  106  IF a=1 THEN f2=2 ELSE f2=3
  107  kx=(c-1)*16+3:ky=(b-1)*16+3
  108  IF ABS(d(b,c))>3 THEN Block1
  109  ON ABS(d(b,c))+1 GOTO Block0,Block4,Block3,Block2
  110  Block0:
  111  f2=0
  112  Block1:
  113  zx=kx:zy=ky:GOSUB Zeichnen
  114  Block2:
  115  zx=kx+6:zy=ky+6:GOSUB Zeichnen
  116  Block3:
  117  zx=kx+6:zy=ky:GOSUB Zeichnen
  118  Block4:
  119  zx=kx:zy=ky+6:GOSUB Zeichnen
  120  RETURN
  121  Zeichnen:
  122  LINE (zx,zy)-(zx+3,zy+3),f2,bf
  123  IF f2=0 THEN LINE (zx,zy)-(zx+4,zy+4),0,bf:RETURN
  124  LINE (zx+4,zy+1)-(zx+4,zy+4),1
  125  LINE (zx+1,zy+4)-(zx+4,zy+4),1
  126  RETURN
  127  Loeschen:
  128  FOR l=5 TO 7
  129    LOCATE l,22:PRINT SPACE$(18)
  130  NEXT l
  131  RETURN
  132  Ende:
  133  WINDOW 1:CLS:END
```

```
  134  Start:
  135  FOR i=1 TO 6:PRINT:NEXT i
  136  COLOR 1,0
  137  PRINT TAB(7)"C R I T I C A L   M A S S"
  138  PRINT TAB(7)"========================="
  139  PRINT:PRINT
  140  PRINT "Ein Taktikspiel für zwei Personen. Für"
  141  PRINT "den Amiga umgeschrieben von A.Widmann"
  142  PRINT
  143  PRINT "Steuerung : Spieler 1 mit dem Joystick"
  144  PRINT "             Spieler 2 mit der Maus"
  145  PRINT
  146  PRINT "  Kennen Sie die Spielregeln (J/N) ?"
  147  Taste2:
  148  a$=INKEY$:IF a$="" THEN Taste2
  149  IF a$="n" THEN Anleitung ELSE RETURN
  150  Anleitung:
  151  CLS:PRINT
  152  PRINT TAB(9)"S P I E L R E G E L N"
  153  PRINT TAB(9)"====================="
  154  PRINT
  155  PRINT " Zwei Spieler setzen abwechselnd einen"
  156  PRINT "roten bzw. blauen Stein auf ein eigenes"
  157  PRINT "oder leeres Feld. Befinden sich auf"
  158  PRINT "einem Feld soviele Steine,wie es Nach-"
  159  PRINT "barfelder hat, so explodiert es. Wenn"
  160  PRINT "ein Feld explodiert, schleudert es alle"
  161  PRINT "Steine auf die Nachbarfelder. Diese und"
  162  PRINT "die gegnerischen Steine darauf gehören"
  163  PRINT "nun zu den eigenen.So können ganze"
  164  PRINT "Kettenreaktionen entstehen."
  165  PRINT
  166  PRINT " Sind nur noch Steine eines Spielers"
  167  PRINT "auf dem Spielfeld vorhanden, so hat"
  168  PRINT "dieser gewonnen."
  169  PRINT
  170  PRINT "Viel Spaß mit Critical Mass !"
  171  PRINT
  172  PRINT "Zum Starten eine Taste drücken !"
  173  taste3:
  174  a$=INKEY$:IF a$="" THEN taste3 ELSE RETURN
```

# Fehlerteufelchen!!!

In Heft 12/87 passierte leider das, was nicht passieren sollte: Im Programm Supermenue hat das Fehlerteufelchen zugeschlagen. Auf Seite 52 befindet sich eine Zeile, die mit 'struct NewWindow windef =' beginnt. Am Ende dieser Zeile befindet sich ein Minuszeichen, welches dort nicht hingehört. Es muß ersatzlos gestrichen werden.

Auch auf Seite 16 ist ein Absatz des Programmes in der Hektik der Herstellung des Magazins unterschlagen worden. Das Listing geht dort bis zur Zeilennummer 146. Fügen Sie bitte nebenstehenden Programmteil ab Zeile Nummer 147 (ohne Zeilennummern) an das Listing von Seite 16 an.

Wir bitten, diese Fehler zu entschuldigen.

```
  136  }
  137  int g,h;
  138     g=sscanf(arv[t+1],"%d",&h);
  139     if (g==-1) {h=-1;t--;}
  140     t++;
  141     return(h);
  142  }
  143
  144
  145  help()
  146  {
  147  printf("*** Echo ***\n");
  148  printf("(C) by RCM\n");
  149  printf("written by G.Glendown\n");
  150  printf("Optionen:\n");
  151  printf("-c        Löscht den Schirm\n");
  152  printf("-e        Löscht von Cursor abwärts\n");
  153  printf("-u/d [n]  um [n] Zeilen hoch/runter scrollen\
n");
  154  printf("-h/r [n]  Cursor [n] Zeilen hoch/runter\n");
  155  printf("-p [x][y] Cursor nach [x][y]\n");
  156  printf("-s[ubip] [a][b]  Style u,b,i,p, Farben [a][b]
\n\n");
  157  exit();
  158  }
  159
  160
```

# Diskinfo

Dieses Programm gibt eine kurze Information über die eingelegte Diskette oder auch RAM-Disk.
Hierfür werden einige Routinen der DOS.Library verwendet und natürlich auch Routinen der Exec.Library (z.B. Openlibrary (-552), damit die Dos.Library überhaupt verwendet werden kann.
Der Aufbau des Programms ist recht einfach und übersichtlich. Zuerst wird die Routine 'Getline' gestartet. Diese bearbeitet die Kommandosequenz (falls eine eingegeben wurde) und sorgt für die jeweiligen Unterprogrammaufrufe.
Die Routine 'Getlock' bestimmt den Zeiger auf den eingegebenen Pfadnamen, der für die nächsten zwei Routinen 'Getinfo' und 'Getexamine' unabdingbar ist.
'Getinfo' und 'Getexamine' besorgen jeweils einige Informationen über die Diskette und tragen diese dann in dem reservierten Speicher 'param' zusammen.
Die drei Unterprogramme 'Diskname', 'Used' und 'Prot' suchen sich aus 'param' ihre Informationen und rufen dann das Unterprogramm 'print' auf, das der Ausgabe bestimmter Texte und Informationen dient. Die Routine 'Ziffer' wird von der Routine 'Used' angesprungen und wandelt Zahlen von Hex in Dez.
'Linefeed' dient zur Ausgabe eines (na, was sonst?) Linefeeds und die Routine 'Help' wird von der 'Getline' Routine aufgerufen, falls ein '?' eingegeben wurde, und gibt lediglich einen Benutzerhinweis aus.

Zur Benutzung des Programms gilt folgendes:
**Diskinfo**: gibt Informatioen über die Diskette in df0: aus
**Diskinfo ?**: zeigt Benutzerhinweise an.
**Diskinfo [Drive]**: gibt Informationen über die jeweilige Diskette oder die RAM-Disk.

Hinweis:
Wer eigene Programme schreiben will, die ebenfalls die Dos-Routinen Info und Examine nutzen, der sollte beachten, daß der Puffer (in diesem Fall 'param') an einer Adresse liegen muß, die durch vier teilbar ist.

```
;-- Assembler-Programm zum Erhalten einer Disk-Information
;-- Autor: Lorenz A.Isbeih

openlibrary = -552
output = -60
lock = -84
info = -114
examine = -102
ioerr = -132
write = -48

execbase = $4

start:
 movem.l d0-d7/a0-a6,-(a7)          ;Register sichern
;--------- uebernehmen der Kommandosequenz ----------
getline:
 lea     name,a1                    ;Zeiger auf Puffer
 subq.l  #2,d0
 blt     getlock                    ;kein Kommando
 cmp.b   #63,(a0)                   ;Vergleich auf '?'
 beq     help                       ; ja: help   nein: weiter
 cmp.l   #4,d0                      ;Laenge der Kommandosequenz
 blt     transfer                   ; < 4: transfer
 move.l  #3,d0                      ; sonst Laenge = 4 (3+1)
;--------- Kommandosequenz in Puffer uebertragen -----
transfer:
 move.b  (a0)+,(a1)+                ;solange uebertragen
 dbra    d0,transfer                ; bis D0 = Null
 move.b  #0,(a1)                    ;mit Null abschliessen
 move.l  d0,a6                      ;Adresse der Dos.Library
;--------- Routine um den Pfad festzulegen ---------
getlock:
 bsr     library
 move.l  #name,d1                   ;Pfadname
 clr.l   d2                         ;Type
 jsr     lock(a6)                   ;Routinenaufruf
 move.l  d0,lockbase                ;Ergebnis sichern
;--------- Information ueber Diskettte holen ------
getinfo:
 move.l  d0,d1                      ;Zeiger auf Pfad
 move.l  #param,d2                  ;Zeiger auf Puffer
 jsr     info(a6)                   ;Info
 tst.l   d0                         ;evtl. Fehler aufgetreten ?
 beq     error                      ; ja: error    nein: weiter
;--------- weitere Informationen holen ------------
getexamine:
 move.l  lockbase,d1                ;Zeiger auf Pfad
 move.l  #param+$20,d2              ;Zeiger auf Puffer
 jsr     examine(a6)                ;Examine
 tst.l   d0                         ;evtl. Fehler aufgetreten ?
 beq     error                      ; ja: error     nein: weiter
;--------- Aufruf der Unterprogramme --------------
 bsr     diskname
 bsr     used
 bsr     prot
 bsr     linefeed
 movem.l (a7)+,d0-d7/a0-a6          ;Register mit alten Werten laden
 rts                                ;zurueck ins CLI
;--------- Ausgabe-Routine ------------------------
print:
 jsr     output(a6)                 ;CLI-Handle ermitteln
 move.l  d0,d1                      ;Handle in D1
 move.l  text,d2                    ;Zeiger auf Text
 move.l  count,d3                   ;Anzahl der auszugebenden Zeichen
 jmp     write(a6)                  ;ausgeben und zurueck ins Hptprg
;--------- Error-Routine --------------------------
error:
 jsr     ioerr(a6)                  ;I/O-Err
 movem.l (a7)+,d0-d7/a0-a6          ;alte Registerinhalte setzten
 rts                                ;zurueck ins CLI
;--------- Disknamen bestimmen und ausgeben -------
diskname:
 move.l  #dname,text                ;Zeiger auf Text
 move.l  #13,count                  ;Anzahl der Zeichen
 bsr     print                      ;Unterprogramm: Ausgabe
 move.l  #param+$28,a1              ;Zeiger auf Disknamen
 move.l  a1,text                    ;Zeiger auf Text
 clr.l   d0                         ;Zaehler loeschen
loop:
 tst.b   (a1)+                      ;Inhalt auf Null ueberpruefen
 beq     ausgabe                    ; ja: ausgabe
 add.l   #1,d0                      ; sonst Zaehler=Zaehler+1
 bra     loop                       ;und das ganze noch mal
ausgabe:
 move.l  d0,count                   ;Anzahl der Zeichen
 bsr     print                      ;Sprung zur Ausgaberoutine
;--------- Routine um ein Linefeed auszugeben -----
linefeed:
 move.l  #return,text               ;Zeiger auf Text
 move.l  #1,count                   ;Anzahl der Zeichen
 bra     print                      ;ausgeben und zurueck
;--------- Routine um Protection zu bestimmen und auszugeben --
prot:
 move.l  #param+$0b,a0              ;Zeiger auf Protection
 move.b  (a0),d0                    ;Inhalt in D0
 tst.b   d0                         ;Inhalt testen
 cmp.b   #$50,d0                    ; ob 50
 bne     prot1                      ; ja: weiter   nein: prot1
 move.l  #protect+$0c,a0 ;Protection = On
 move.b  #$20,(a0)+
 move.b  #$4f,(a0)+
 move.b  #$6e,(a0)+
 move.b  #$20,(a0)
 jmp     prot2
;--------- Routine wird nur angesprungen,wenn Protection = Off -
prot1:
 move.l  #protect+$0d,a0            ;Protection  = Off
 move.b  #$4f,(a0)+
 move.b  #$66,(a0)+
 move.b  #$66,(a0)
;--------- Protection ausgeben --------------------
prot2:
 move.l  #protect,text              ;Zeiger auf Text
 move.l  #18,count                  ;Anzahl der Zeichen
 bra     print                      ;ausgeben und zurueck
;--------- Routine um die Belegten Bloecke zu bestimmen -----
used:
 move.l  #param+$12,a2              ;Zeiger auf Used-Blocks
 move.w  (a2),d1                    ;Inhalt in d1
 move.l  #block+$0d,a4              ;Zeiger auf Puffer
 move.l  #1000,d2
;--------- Hex => Dez wandeln --------------------
ziffer:
```

```
 divu    d2,d1                ;D1/D2 (1000,100,10,1)
 add     #$30,d1              ;Ergebnis + $30 ( => Ascii )
 move.b  d1,(a4)+             ;und in Puffer schreiben
 clr     d1                   ;untere Word loeschen
 swap    d1                   ;und jetzt der Rest vom Ergebnis
 divs    #10,d2               ;Inhalt von D2 durch 10 teilen
 bne     ziffer               ;solange <> Null : Ziffer
 move.l  #block,text          ;Zeiger auf Text
 move.l  #19,count            ;Anzahl der Zeichen
 bra     print                ;ausgeben und zurueck
;--------- Routine zum Oeffnen der Dos.Library ------
library:
 move.l  execbase,a6          ;Adresse der Exec.Library
 lea     dosname,a1           ;Zeiger auf LibName
 clr.l   d0                   ;Version: egal
 jsr     openlibrary(a6)      ;und oeffnen
 move.l  d0,a6                ;Adresse der Dos.Library sichern
 rts                          ;und zurueck ins Hauptprogramm
;--------- Ausgabe einer Bedienungsanweisung ---------
help:
 bsr     library              ;Library oeffnen
 move.l  #usage,text          ;Zeiger auf Text
 move.l  #26,count            ;Anzahl der Zeichen
 bsr     print                ;und ausgeben
 movem.l (a7)+,d0-d7/a0-a6    ;Register laden
 rts                          ;und zurueck ins CLI

lockbase: dc.l 0
text:     dc.l 0
count:    dc.l 0
dosname:  dc.b 'dos.library',0
name:     dc.b 'df0:',0
dname:    dc.b 'DiskName   : ',0
block:    dc.b 'Used-Blocks:      ',$0a,0
protect:  dc.b 'Protection :     ',$0a,0,0
return:   dc.b $0a
param:    blk.b $50,0              ;Puffer fuer Disk-Daten
usage:    dc.b 'Usage: DiskInfo [Drive]',$0a,$0a,0
```

# DisplayAlert

Das Herzstück dieses Programms ist die Intuition-Routine 'Display Alert'. Mit dieser Routine läßt sich auf einfachste Weise ein eigener 'Guru a la carte' auf den Bildschirm bringen.
Des weiteren läßt sich sogar noch abfragen, welche Maustaste zur Beendigung gedrückt wurde und dann, je nach Ausbau des Programms, kann ein ganz bestimmtes Unterprogramm aufgerufen werden, genauso wie es bei den richtigen Guru-Meldungen der Fall ist.
Mal ganz von der Funktion abgesehen, ist es verblüffend, mit welch einfachen Mitteln man schon so viel erreichen kann.
Dieses Programm ist sogar geeignet, als Demo in einer Startup-Sequence verwendet zu werden.

```
;-- Assembler-Programm zum Aufrufen von Alerts unter Intuition
;-- Autor: Lorenz A.Isbeih

DisplayAlert = -90
OpenLibrary  = -408
CloseLibrary = -414

execbase = 4

start:
;------- intuition.library laden -------
 move.l execbase,a6           ;Adresse der Exec.library
 lea    intname,a1            ;Zeiger auf Library Namen
 jsr    OpenLibrary(a6)       ;Routinenaufruf
 move.l d0,intbase            ;Adresse sichern
;------- Alert aufrufen -------
 move.l d0,a6                 ;Adresse der Intuition.Library
 move.l #$0,d0                ;Alert-Nummer
 lea    string,a0             ;Zeiger auf Text-Struktur
 move.l heigth,d1             ;Hoehe
 jsr    DisplayAlert(a6)      ;Routinenaufruf
 cmp.b  #$01,d0               ;Vergleich,wenn d0
 beq    left                  ; = $1 => linke Maustaste gedrueckt
 bne    right                 ; = $0 => rechte Maustaste gedrueckt
ende:
;------- Schliessen der Intuition.library -------
 move.l execbase,a6           ;Adresse der Exec.library
 move.l intbase,a1            ;Adresse der Intuition.library
 jsr    CloseLibrary(a6)      ;Routinenaufruf
;------- Zurueck ins CLI -------
 rts

;------- Hier koennten Ihre Routinen stehen -------
left:
 move.l  #5,d3
 jmp     ende
right:
 move.l  #6,d3
 jmp     ende

;------- Variablendeklaration -------
intname: dc.b 'intuition.library',0
even
intbase: dc.l 0
heigth:  dc.l 40
string:
 dc.b $0,$d2,$0b,'This is your own Alert',0,1
 dc.b $0,$7b,$15,'A little demo to show you how to use Alerts',0,1
 dc.b $0,$aa,$21,'<Press Mouse-Button to continue>',0,0
```

# ChangeName

Dieses Programm soll kein sinnvolles Utility darstellen, sondern ein wenig über die Strukturen der Intuition.Library und des CLI-Windows aufklären.
Was macht das Programm?
Es empfängt bei Aufruf die Kommandosequenz und schreibt diese dann in die Kopfzeile des Windows.
Hierzu muß der Benutzer jedoch das Window einmanl inaktivieren und dann wieder aktivieren, damit der Name übernommen wird.
Der Trick des ganzen ist; man sucht sich die Adresse des aktiven Windows aus der Intuition.library (Langwort ab dem $34. Byte) und sucht wiederum die Adresse des Window-Titels und überträgt nun einfach die erhaltene Kommandosequenz in den Puffer ab dieser Adresse.
Zur Benutzung: Es braucht nicht unbedingt eine Kommandosequenz eingegeben werden. Falls diese fehlt, so wird der Balken geleert.

```
;-- Assembler-Programm zum Aendern des Cli-Window-Titel
;-- Autor: Lorenz A.Isbeih

openlibrary = -552
output      = -60
write       = -48

execbase    = $4

;--------- uebernehmen der Kommandosequenz --------
getline:
 lea     linebuffer,a1        ;Zeiger auf Puffer
 subq.l  #2,d0
 blt     change               ;kein Kommando
 cmp.b   #63,(a0)             ;Vergleich auf '?'
 beq     help                 ;ja: help
 cmp.l   #72,d0               ;ueberpruefen der Laenge
 blt     transfer             ;wenn kleiner,dann uebertragen
 move.l  #72,d0               ;sonst max. 73 (72+1) Zeichen
;--------- Kommandosequenz in Puffer uebertragen --
transfer:
 move.b  (a0)+,(a1)+          ;uebertragen
 dbra    d0,transfer          ; bis D0 < Null
 move.b  #0,(a1)              ;und mit Null abschliessen
;--------- Titel aendern -------------------------
change:
 lea     intname,a1           ;Zeiger auf LibName
 jsr     library              ;Intuition.Library oeffnen
 move.l  d0,a1                ;Zeiger auf Intuition.Library
 move.l  $34(a1),a1           ;Adresse des CLI-Windows
 move.l  $20(a1),a1           ;Zeiger auf Fenster-Titel
 lea     linebuffer,a2
```

```
;--------- aendern des eigegebenen Namens in Titel
loop:
 move.l  (a2)+,(a1)+          ;solange uebertragen
 bne     loop                 ; bis (a2) = Null
 rts
;--------- Routine zum oeffnen von Libraries -----
library:
 move.l  execbase,a6          ;Adresse der Exec.Library
 clr.l   d0                   ;Version: egal
 jmp     openlibrary(a6)      ;oeffnen und zurueck zum Hauptprogramm
;--------- Hilfestellung zur Programmbenutzung ---
help:
 lea     dosname,a1           ;Zeiger auf LibName
 jsr     library              ;Dos.Library oeffnen
 move.l  d0,a6                ;Adresse der Dos.Library
 jsr     output(a6)           ;CLI-Handle ermitteln
 move.l  d0,d1                ;Handle in D1
 move.l  #usage,d2            ;Zeiger auf Text
 move.l  #26,d3               ;Anzahl der auszugebenden Zeichen
 jmp     write(a6)            ;ausgeben und ins Hauptprogramm

linebuffer: blk.b 74,0        ;Puffer fuer Kommandosequenz reservieren
usage:      dc.b 'Usage: ChangeName [name]',$0a,$0a,0
intname:    dc.b 'intuition.library',0
dosname:    dc.b 'dos.library',0
```

# DigiSound

Zu diesem Listing gibt es eigentlich nicht viel zu sagen, weil alles, was zu sagen wäre, schon im Listing selbst steht. Wer jedoch ein wenig experimentieren will, der sollte sich folgende Information ansehen:

| | | | |
|---|---|---|---|
| Audiochannel 0 | $dff0a0 | Length | Channel + $04 |
| Audiochannel 1 | $dff0b0 | Period | Channel + $06 |
| Audiochannel 2 | $dff0c0 | Volume | Channel + $08 |
| Audiochannel 3 | $dff0d0 | Data | Channel + $0a |

DMA-Control-Register $dff096

| Bit: | 15 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|
| | set/clr | Channel 3 | Channel 2 | Channel 1 | Channel 0 |

```
;-- Assembler-Programm zum Abspielen von digitalen Sounds
;--  Anweisung :
;--         - assemblieren
;--         - mit 'ri' ein Soundstueck nach 'data' laden
;--         - die Laenge (= der Filelaenge)
;--         - die Geschwindigkeit einstellen (probieren)
;--         - mit 'j start' starten
;--         - mit 'j soundoff' beenden
;-- Viel Spass        Lorenz A.Isbeih

laenge = 65536*2              ;Laenge (max. 65536*2)
rate = 300                    ;Geschwindigkeit
vol = 64                      ;Lautstaerke 1-64

start:
 move.l  #$dff0a0,a0          ;Adresse fuer Kanal 0 (0-3)
soundon:
 move.l  #data,(a0)+          ;Zeiger auf Sounddaten
 move.w  #laenge/2,(a0)+      ;Laenge
 move.w  #rate,(a0)+          ;Geschwindigkeit
 move.w  #vol,(a0)            ;Lautstaerke
 move.w  #$820f,$dff096       ;alle 4 Kanaele an (DMA-Control)
 rts

soundoff:
 move.w  #$f,$dff096          ;alle 4 Kanaele abschalten
 rts

data:                         ;ab dieser Adresse stehen die Daten
```

# OwnScreen

Neben den Informationen, die bereits im Listing stehen, noch einige Anmerkungen:
Die Anzahl der Bitplanes ist einerseits bestimmt durch den Darstellungsmodus und gibt andererseits die Anzahl der darstellbaren Farben an. Man muß jedoch beachten, daß eine Screen mit mehreren Bitplanes auch mehr Speicherplatz einnimmt.

Dazu folgendes:

| Auflösung | Bytes pro Bitplane | Anzahl der Farben |
|---|---|---|
| 320 x 256 | 10240 | |
| 320 x 512 | 20480 | |
| 640 x 256 | 20480 | $2^{\text{Anzahl der Bitplanes}}$ |
| 640 x 512 | 40960 | |

| Darstellungsmodi: | | Screentypen | |
|---|---|---|---|
| Genlock | $02 | Workbenchscreen | $01 |
| Interlace | $04 | Customscreen | $0f |
| PFBA | $40 | ShowTitle | $10 |
| HalfBrite | $80 | Deeping | $20 |
| Genlock-Audio | $100 | CustomBitmap | $40 |
| ColorOn | $200 | ScreenBehind | $80 |
| DBPLF | $400 | ScreenQuiet | $100 |
| HAM | $800 | | |
| VP_Hide | $2000 | | |
| Sprites | $4000 | | |
| Mode_640 | $8000 | | |

```
;-- Assembler-Programm zum Oeffnen von Screens unter Intuition
;-- Autor: Lorenz A.Isbeih

openlibrary = -552
openscreen  = -198
closescreen = -66

execbase    = $4

start:
 library:
  move.l execbase,a6          ;Adresse der Exec.Library
  lea    intname,a1           ;Zeiger auf LibName
  clr.l  d0                   ;Version: egal
  jsr    openlibrary(a6)      ;und oeffnen
  move.l d0,intbase           ;Adresse der Intuition.Library sichern
 screenon:
  move.l intbase,a6
  lea    screen_defs,a0       ;Zeiger auf Def.-Tabelle
  jsr    openscreen(a6)       ;und oeffnen
  move.l d0,scrbase           ;Adresse des Screens sichern
 wait:
  btst     #10,$dff016        ;rechte Maustaste gedrueckt ?
  bne    wait                 ;nein = wait   ja = weiter
 screenoff:
  move.l intbase,a6
  move.l scrbase,a0           ;Adresse des Screens
  jsr    closescreen(a6)      ;und schliessen
 ende:
  rts                         ;zurueck ins CLI oder Seka

intname: dc.b 'intuition.library',0
even
intbase: dc.l 0
scrbase: dc.l 0
screen_defs:
  dc.w  0,0,320,200           ;x/y-Position u. Breite/Hoehe
  dc.w  2                     ;Bitplanes
  dc.b  1,3                   ;Textfarbe/Hintergrungfarbe
  dc.w  2                     ;Darstellungs-Modus
  dc.w  15                    ;Screentyp
  dc.l  0                     ;Font
  dc.l  titel                 ;Titel
  dc.l  0                     ;Gadgets
  dc.l  0                     ;Bitmap

titel: dc.b "Roeske's Computer Mag.",0
```

SCULPT 3D
incl.dt.Handbuch DM 229,-

SILVER DM 298,-

Ray Tracer's
ANIMATE 3D DM call

FORMS IN FLIGHT DM 169,-

APPRENTICE: Animation
incl.dt.Hb.+VHS band
DM 698,-
2 Demodisks DM 10,-

VIDEOSCAPE 3D
incl.dt.Handbuch DM 398.-
1 Demodisk DM 5,-

PERFECT SOUND
stereo Digitizer
für alle AMIGA DM 225,-

ECE MIDIINTERFACE
für alle AMIGA DM 139,-

DIGI VIEW incl.dt.Hb DM 398,-

Gender Changer
Adapter zum Digi View
für A500/A2000 DM 48,-

R G B Splitter
zB.f.Digi View DM 598,-

DIGI PAINT DM 139,-
BUTCHER DM 79,-

POLAROID PALETTE
m.IMRINT f.AMIGA DM 6750,-

Ihre AMIGA Bilder auf DIA
INFO am Telefon

GENLOCK 8700
für alle AMIGA DM 1095,-

STUDIO MAGIC
Digitales Sound Studio

ideal zum Perfect Sound
Stereo Digitizer

DM 129,-

PRO VIDEO CG I DM 498,-

FONT SET I DM 249,-
FONT SET II DM 249,-

PRO Video Demodisk DM 5,-

2 MB Golem Box
für A500/A1000 DM 998,-

64 EMULATOR
mit seriellem Interface
Bitte AMIGA Typ angeben!
DM 149,-

TEST DRIVE DM 79,-

VTC-1000 VHS-Timecode
Schnittsteuereinheit
incl.Playermodifikation
DM 3798,-

ACQUISITION
Dateiverwaltung DM 528,-

video

deutsche Handbücher für:
Appr.:Animator DM 79,95
Videoscape 3D DM 39,95
Sculpt 3D DM 39,95
dt.Hb.in Vorbereitung:
Aztec C DM 98,-
Silver DM call,
Ray Tracer's Animate 3D

video LOFT film
HARD & SOFTware GmbH
Fiedlerstr.22-32
D-3500 KASSEL

GmbH
ph.: 0561 - 87 79 28
87 33 99
fax: 0561 - 87 80 48

Anrufen und
L O F T - P O S T
schicken lassen! Neueste
D E M O D I S K S
Stck. DM 5,-

im Brief (Schein o. Marken) o. KtNr.61655
Stadtsparkasse Kassel
BLZ 520 501 51
o.NN + DM 6.- NN Geb.

gesetzt mit PAGESETTER (DM 349,-) auf A 500 L.H.87

# WorkbenchSize

Daß der Amiga riesige Bildschirme ansteuern kann, ist ja schon bekannt, Aber daß man die Workbench auch größer als die normalen 640*256 Pixels machen kann, ist kaum bekannt und auch nirgends dokumentiert.
Das Programm WorkbenchSize ist dazu ausgelegt, eine beliebige bootbare Diskette so zu verändern, daß man danach eine Workbench von bis zu 767*383 Pixels zur Verfügung hat. Dabei wird nur mit Maus gearbeitet; alle Abfragen erfolgen über Gadgets.
Zum Programm: WorkbenchSize überprüft zuerst die angeschlossenen Drives, so daß nicht aus Versehen auf nicht vorhandene Laufwerke zugegriffen wird, was meißt diesen nervenden Requester links oben bewirkt...
Danach werden alle Laufwerk-Select Gadgets deaktiviert, deren entsprechender Drive nicht angeschlossen ist.
Hat der Benutzer sich entschieden, in welchem Laufwerk sich die zu SIZEnde Diskette befindet, wird das benötigte File von der Diskette eingelesen.
Daraufhin erscheint auf dem Bildschirm ein Window, das die aktuelle eingestellte Größe der Screen darstellt, desweiteren befinden sich in dem Window noch die Angaben zur X- und Y-Ausdehnung und ein 'OK'-Gadget. Nun kann man mittels des Window-Size-Gadgets die Größe verändern, was man durch drücken des 'OK'-Gadgets quittiert. Danach wird das neue File auf Disk geschrieben.
Man kann jederzeit das Programm abbrechen, indem man irgendeines der Close-Gadgets drückt.
Zur Theorie: WorkbenchSize nutzt aus, daß in dem Preferences-File (devs/system-configuration) noch einige Daten vorhanden sind, die nicht mit Preferences beeinflußbar sind. Dazu gehören nämlich jene Werte, die den Betrag der Vergrößerung enthalten. Diese beiden Werte können in dem bereich von 0 bis 127 liegen, wobei sie jeweils zu der 'normalen' Anzahl von 640 bzw. 256 dazugezählt werden.
Natürlich ist es nicht unbedingt sinnvoll, die Screen wirklich auf 767*383 sizen sollte. Im Normalfall dürfte eine Vergrößerung in X um 32 Pixels und in Y um 20 Pixels ungefähr so bemessen sein, daß nach Verschieben des Screen alles sichtbar ist.

Das Programm wurde unter Aztec C V3.2 geschrieben und sollte auch mit V3.4 ordentlich laufen.

```
/*       Workbench Sizing
     (C) 1987 by Roeske's Computer Mag.
     written 1987 by G.Glendown
                                        */

/* Includes */
#include "stdio.h"
#include "intuition/intuition.h"
#include "graphics/gfxbase.h"
#include "exec/types.h"
#include "devices/trackdisk.h"
#include "exec/devices.h"
#include "exec/io.h"
#include "libraries/dos.h"

/* Globale Def's */
#define ITextList IText5
#define NL NULL
#define ITixtList IText3
#define SIZE_OUTX 500
#define SIZE_OUTY 110
#define ever (;;)
#define UBP (UBYTE *)


struct IntuiMessage *msg,*GetMsg();
struct IntuitionBase    *IntuitionBase;
struct GfxBase          *GfxBase;
struct Screen  *scr, *OpenScreen();
struct Window  *win,*win1,*win2, *OpenWindow();

long lncnt;
struct TextAttr TOPAZ60 = {(STRPTR)"topaz.font",TOPAZ_SIXTY,0,0};
 char inp[]={"df0:\0"};
struct Gadget *gadgadr;

int          checkdisk();
LONG         OpenDevice();
VOID         CloseDevice(),DeleteStdIO(),DeletePort()

struct       IOStdReq *ior,*CreateStdIO();
struct       MsgPort *CreatePort(),*mp;


USHORT BordirVectors1[] = {0,0,49,0,49,14,0,14,0,0};
USHORT BordirVectors2[] = {0,0,49,0,49,14,0,14,0,0};
USHORT BordirVectors3[] = {0,0,49,0,49,14,0,14,0,0};
USHORT BordirVectors4[] = {0,0,49,0,49,14,0,14,0,0};

struct Border Bordir1 = {-2,-1,2,0,JAM1,5,(SHORT *)BordirVectors1,NL};
struct Border Bordir2 = {-2,-1,2,0,JAM1,5,(SHORT *)BordirVectors2,NL};
struct Border Bordir3 = {-2,-1,2,0,JAM1,5,(SHORT *)BordirVectors3,NL};
struct Border Bordir4 = {-2,-1,2,0,JAM1,5,(SHORT *)BordirVectors4,NL};

struct IntuiText IText5 = {1,0,JAM1,4,22,&TOPAZ60,(UBYTE *)"Bitte Laufwerk angeben:",NL};
struct IntuiText IText4 = {1,0,JAM1,163,3,&TOPAZ60,(UBYTE *)"DF3:",NL};
struct IntuiText IText3 = {1,0,JAM1,110,3,&TOPAZ60,(UBYTE *)"DF2:",&IText4};
struct IntuiText IText2 = {1,0,JAM1,57,3,&TOPAZ60,(UBYTE *)"DF1:",&IText3};
struct IntuiText IText1 = {1,0,JAM1,3,3,&TOPAZ60,(UBYTE *)"DF0:",&IText2};

struct Gadget df3 = {NL,167,37,46,13,GADGHCOMP,RELVERIFY,BOOLGADGET,
     (APTR)&Bordir1,NL,NL,0,NL,0x33,NL};
struct Gadget df2 = {&df3,114,37,46,13,GADGHCOMP,RELVERIFY,BOOLGADGET,
     (APTR)&Bordir2,NL,NL,0,NL,0x32,NL};
struct Gadget df1 = {&df2,61,37,46,13,GADGHCOMP,RELVERIFY,BOOLGADGET,
     (APTR)&Bordir3,NL,NL,0,NL,0x31,NL};
struct Gadget df0 = {&df1,8,37,46,13,GADGHCOMP,RELVERIFY,BOOLGADGET,
     (APTR)&Bordir4,NL,&IText1,0,NL,0x30,NL};

struct NewWindow windef1 = {81,34,262,63,0,1,
     GADGETDOWN | GADGETUP | CLOSEWINDOW,
     WINDOWDRAG | WINDOWCLOSE | SIZEBBOTTOM | ACTIVATE | RMBTRAP | NOCAREREFRESH,
     &df0,NL,(UBYTE *)"WB SIZING by G.Glendown",NL,NL,0,0,0,0,WBENCHSCREEN};

struct IntuiText ITixt2 = {1,0,JAM1,3,14,&TOPAZ60,(UBYTE *)"K",NL};
struct IntuiText ITixt1 = {1,0,JAM1,3,4,&TOPAZ60,(UBYTE *)"0",&ITixt2};

USHORT BorderVectors1[] = {0,0,20,0,20,26,0,26,0,0};

struct Border Border1 = {-2,-1,1,0,JAM1,5,(SHORT *)BorderVectors1,NL};

struct Gadget Gadget4 = {NL,91,13,17,25,GADGHCOMP,RELVERIFY,BOOLGADGET,
     (APTR)&Border1,NL,&ITixt1,0,NL,71,NL};

struct NewWindow windef2 = {150,52,128,51,0,1,
     GADGETDOWN|GADGETUP|CLOSEWINDOW,
     WINDOWSIZING|WINDOWDRAG|WINDOWCLOSE|SIZEBRIGHT|ACTIVATE|NOCAREREFRESH,
```

```
    &Gadget4,NL,(UBYTE *)"WB SIZING by G.Glendown",NL,
NL,128,51,255,178,WBENCHSCREEN};

struct NewWindow windef = {0,0,SIZE_OUTX,SIZE_OUTY,0,
1,CLOSEWINDOW,WINDOWDEPTH|WINDOWDRAG|WINDOWCLOSE|NOCAREREFRE
SH,
    NL,NL,(UBYTE *)" WB SIZING (C) 1987 by G.Glendown"
,NL,NL,0,0,0,0,WBENCHSCREEN};

FILE *fopen(),*fp;

char name[] = {"devs/system-configuration"};
char dr1[]={"df1:"},dr2[]={"df2:"},dr3[]={"df3:"};

int xs,ys,xc,yc;

char dumm[4];
    struct Process *OurTask;
    struct Window *old_pr_WindowPtr;

main()
{
int n;
char c;
char fipo[],firo[];
int w,l,ws[120],i;
register struct RastPort *rp;

    openstuff();
    rp = & (scr -> RastPort);

    if (!(win = OpenWindow (&windef))) {
              CloseLibrary(IntuitionBase);
              CloseLibrary(GfxBase);
              exit (FALSE);
    }

    lncnt=1;

    printw (win,"**************************************
**********************");
    printw (win,"****           Workbench Sizing by Ga
y  Glendown      ****");
    printw (win,"**************************************
**********************");
    printw (win,"Version 3.0        (C) 1987 by Roeske
 Computer Mag.");
    printw (win,"");
    printw (win,"Nach dem Benutzen dieses Programmes R
et durchfuehren und");
    printw (win,"mit der gesizeten Disk booten, dann m
tels 'Preferences'");
    printw (win,"die Screen optimal verschieben.");

    fp=NULL;

    while (fp==NULL) {

              getdrive();
              strcpy(fipo,inp);
              strcat(fipo,name);strcpy(firo,fipo);
              fp=fopen(firo,"r");
              l=0;

              if (fp!=NULL) {

                        while (l<117) {
                                  w=getw(fp);
                        l++;
                                  ws[l]=w;
                        }

                        fclose(fp);
                        yc=(int)(ws[109]/256);
                        xc=ws[109]%256;
                        windef2.Width=(SHORT)(xc+128);
                        windef2.Height=(SHORT)(yc+51);
              }

              if (fp==NULL) printw (win,"Keine zulaess
ige Boot-Disk!!!");
    }

    getsize();
    ws[109]=yc*256+xc;
    fp=fopen(firo,"w");

    if (fp!=NULL) {

              for (l=1;l<117;l++) {
                        i=putw((int)ws[l],fp);
              }

              fclose(fp);
    }

    printw (win,"Viel Vergnuegen mit ihrer neuen RIESE
N-Workbench Screen");
    printw (win,"                                    Gar
ry Glendown");

    WaitPort (win -> UserPort);
    closestuff ();
    exit (TRUE);
}

getsize()

{
int flag,class,code,num;
register struct RastPort *rp;

    if (!(win2 = OpenWindow (&windef2))) {
     CloseLibrary(IntuitionBase);
     CloseLibrary(GfxBase);
     CloseWindow(win);
          exit (FALSE);
    }

    rp = win2 -> RPort;
    SetDrMd (rp,JAM2);
    SetAPen(rp,3L);
    Move (rp,6L,22L);
    Text (rp,(UBYTE *)"New Size ?",10L);
    Move (rp,6L,32L);
    Text (rp,(UBYTE *)"X:",2L);
    Move (rp,6L,42L);
    Text (rp,(UBYTE *)"Y:",2L);
    SetAPen(rp,1L);
    xs=(int)(win2->Width)-128+640;
    ys=(int)(win2->Height)-51+256;
    sprintf(dumm,"%3d",xs);
    Move (rp,26L,32L);
    Text(rp,dumm,(long)(strlen(dumm)));
    Move (rp,26L,42L);
    sprintf(dumm,"%3d",ys);
    Text(rp,dumm,(long)(strlen(dumm)));

    for (flag=0;flag==0;) {
              xs=(int)(win2->Width)-128+640;
              ys=(int)(win2->Height)-51+256;
              sprintf(dumm,"%3d",xs);
              Move (rp,26L,32L);
              Text(rp,dumm,(long)(strlen(dumm)));
              Move (rp,26L,42L);
              sprintf(dumm,"%3d",ys);
              Text(rp,dumm,(long)(strlen(dumm)));

              while ((msg=(struct IntuiMessage *)GetMs
g(win2->UserPort))&&flag==0) {
                        class=msg->Class;
                        code=msg->Code;
                        if (class==CLOSEWINDOW) {CloseW
indow(win2);closestuff();exit(TRUE);}

                        if (class==GADGETUP) {
                                  gadgadr=(struct Gadge
t *)msg->IAddress;

                                  if (gadgadr->GadgetID
==71) {
                                            xc=xs-640;
                                            yc=ys-256;
                                            ReplyMsg(msg
```

```
);
  234                                          CloseWindow(
win2);
  235                                          flag=1;
  236                                  }
  237
  238                          }
  239
  240                          if (flag!=1) ReplyMsg(msg);
  241                }
  242
  243                while (msg=(struct IntuiMessage *)GetMsg
(win->UserPort)) {
  244
  245                          if ((msg->Class==CLOSEWINDOW)&&
(flag==0)) {
  246                                       ReplyMsg(msg);CloseWi
ndow (win2);closestuff();exit(TRUE);
  247                          }
  248
  249                          ReplyMsg(msg);
  250                }
  251     }
  252  }
  253
  254  getdrive()
  255
  256  {
  257  int flag,class,code,num;
  258  register struct RastPort *rp;
  259
  260     if (!(win1 = OpenWindow (&windef1))) {
  261      CloseLibrary(IntuitionBase);
  262      CloseLibrary(GfxBase);
  263      CloseWindow(win);
  264           exit (FALSE);
  265     }
  266
  267     rp = win1 -> RPort;
  268     PrintIText (rp,&ITextList,0L,0L);
  269     checkdrives();
  270
  271     for (num=-1;num==-1;) {
  272     while (msg=(struct IntuiMessage *)GetMsg(win1->Use
rPort)) {
  273                class=msg->Class;
  274                code=msg->Code;
  275                ReplyMsg(msg);
  276                if (class==CLOSEWINDOW) {CloseWindow(win
1);closestuff();exit(TRUE);}
  277                if (class==GADGETUP) {
  278                          gadgadr=(struct Gadget *)msg->I
Address;
  279                          num=gadgadr->GadgetID;
  280                          if ((num<0x30)||(num>0x33)) num
=-1;
  281                }
  282     }
  283
  284     while (msg=(struct IntuiMessage *)GetMsg(win->User
Port)) {
  285                if (msg->Class==CLOSEWINDOW) {CloseWindo
w (win1);closestuff();exit(TRUE);}
  286     }
  287     }
  288     inp[2]=(char)num;
  289     CloseWindow(win1);
  290  }
  291
  292  checkdrives()
  293
  294  {
  295  int a;
  296     a=checkdisk(1);
  297     if (a==0) OffGadget(&df1,win1,NULL);
  298     a=checkdisk(2);
  299     if (a==0) OffGadget(&df2,win1,NULL);
  300     a=checkdisk(3);
  301     if (a==0) OffGadget(&df3,win1,NULL);
  302  }
  303
  304  int checkdisk(t)
  305
  306  int t;
  307
  308  {
  309  int a=0;
  310     if (mp=CreatePort(NULL,0L)) {
  311               if (ior=CreateStdIO(mp)) {
  312                         if (!OpenDevice(TD_NAME,(long)t
,ior,0L)) {a=1;CloseDevice(ior);}
  313                         DeleteStdIO(ior);
  314               }
  315               DeletePort(mp);
  316     }
  317     return (a);
  318  }
  319
  320
  321  openstuff ()
  322
  323  {
  324     char *OpenLibrary();
  325
  326     if (!(IntuitionBase = (struct IntuitionBase *)
  327        OpenLibrary ("intuition.library", 1L))) {
  328           exit (FALSE);
  329     }
  330
  331     if (!(GfxBase = (struct GfxBase *)
  332        OpenLibrary ("graphics.library", 1L))) {
  333           CloseLibrary(IntuitionBase);
  334      exit (FALSE);
  335     }
  336  }
  337
  338  closestuff ()
  339
  340  {
  341     CloseWindow (win);
  342     CloseLibrary (GfxBase);
  343     CloseLibrary (IntuitionBase);
  344  }
  345
  346
  347  printw(win,text)
  348
  349  char text[];
  350  struct Window *win;
  351
  352  {
  353  int flag,code;
  354  register struct RastPort *rp;
  355     rp = win -> RPort;
  356
  357     if (lncnt==11) {
  358               lncnt--;
  359               ScrollRaster(rp,0L,9L,3L,11L,(long)(SIZE
_OUTX-3),(long)(SIZE_OUTY-3));
  360     }
  361     SetAPen(rp,1L);
  362     Move(rp,5L,(long)(lncnt*9+9));
  363     Text(rp,text,(long)(strlen(text)));
  364     lncnt++;
  365  }
  366
```

# Ein erweiterter Bobeditor

Jedem Commodore Amiga liegt eine Basic-Diskette bei, auf der sich einige nützliche Demos und Utilities befinden. Eines davon ist der Bobeditor. Wir haben diesen etwas erweitert und drucken ihn hier ab.

Nach dem Starten des Programms befinden sich auf dem Bildschirm ein Zeichenfeld, ein Band mit den aktuellen 32 Farben, eine Statuszeile und die RGB-Werte der momentanen Zeichenfarbe. Durch Anklicken auf ein Plus-Zeichen, können diese Farbwerte geändert werden.

Wenn Sie mit der Maus eine Farbe des Farbbandes anklicken, so wird diese zur aktuellen Zeichenfarbe.

Die Größe des Bobs können Sie einstellen, indem Sie das kleine Quadrat rechts unterhalb des Zeichenfeldes anklicken, festhalten und verschieben.

Mit der rechten Maustaste lassen sich die Menüpunkte sichtbar machen. Es gibt unter dem Tools-Menue Pen, BigPen, womit schmale oder breite Linien gezeichnet werden können.

Auch Line funktioniert wie im alten Programm. Circle arbeitet nun so, daß durch Anklicken der Mittelpunkt (nicht wie vorher die linke obere Ecke) des Kreises bzw. der Ellipse festgelegt wird und durch Ziehen mit der Maus der vertikale und horizontale Radius eingestellt werden können.

Square zieht Rechtecke und Quadrate wie gehabt und Erase radiert mit breitem Radiergummi.

Fill wurde wieder geändert. Es füllt nun eine Fläche und stellt danach Pen ein. Dies wurde umgestellt, weil sich in der Praxis gezeigt hat, daß häufig diese Funktion angewählt wurde, ohne daß diese gewünscht wurde.

Es gibt einen Menueheader namens Test. Mit dem können Bobs in der Bewegung ausgetestet werden. Diese müssen auf Diskette vorliegen und mit dem Bobeditor abgespeichert worden sein.

Das Laden und Speichern geschieht wie in den meisten Programmen (wie in der Originalversion). Es wird aber im Gegensatz zu dieser ein File mit der Farbpalette angelegt. So können Sie Bobs und Farben getrennt laden und in Ihre Programme einbauen.

Dieser Bobeditor wird im Laufe der nächsten Wochen weiter verbessert; u.a. wollen wir noch Scrolling, AND/OR/XOR verschiedener Bilder, Animationen, usw. einfügen.

```
    1    CLEAR,40000&,1024
    2
    3    DEFINT a-z
    4    DIM Col(31,2)
    5
    6
    7        collisionPlaneIncluded=2
    8        imageShadowIncluded=4
    9        SAVEBACK=8
   10        OVERLAY=16
   11        SAVEBOB=32
   12
   13    DEF FNArraySize& = 3+INT((BobRight+16)/16)*(BobBott
om+1)*Depth
   14    DIM DrawRect(3),ToolName$(7)
   15
   16    LIBRARY "graphics.library"
   17
   18
   19
   20    WinY=200: WinX=290
   21    Depth=5
   22    scrn=1
   23    SCREEN scrn,320,220,Depth,1
   24    WINDOW 1,"RCM-Bobeditor 1",(0,0)-(WinX,WinY),31,scr
n
   25    sWindow& = WINDOW(7)
   26    sScreen& = PEEKL(sWindow&+46)
   27    sViewPort& = sScreen&+44
   28    sColorMap& = PEEKL(sViewPort&+4)
   29    colorTab& = PEEKL(sColorMap&+4)
   30    FOR i=0 TO 31
   31      CT=PEEKW(colorTab&+(2*i))
   32      FOR j=2 TO 0 STEP -1
   33        Col(i,2-j)=INT(CT/16^j):CT=CT-Col(i,2-j)*16^j
   34      NEXT
   35    NEXT
   36
   37    GOSUB InitConstant
   38    GOSUB InitFile
   39    GOSUB InitMenu
   40
   41  StartOver:
   42    ON MENU GOSUB CheckMenu : MENU ON
   43    ON MOUSE GOSUB CheckMouse : MOUSE ON
   44    ON BREAK GOSUB IgnoreBreak: BREAK ON
   45    DrawBoundary
   46    GOSUB PrintStatus
   47    Unfinished = -1
   48    WHILE Unfinished
   49      SLEEP
   50    WEND
   51
   52    MENU OFF: MOUSE OFF
   53    SCREEN CLOSE 1
   54    WINDOW CLOSE 1
   55    WINDOW 1,,,,-1
   56
   57    MENU RESET
   58    CLS
   59    END
   60
   61  InitConstant:
   62    MaxTool=7
   63    ToolMode=1
   64    CurrentColor=1
   65    MaxY=120: MaxX=500
   66    MaxY10=MaxY+10: MaxX10=MaxX+10
   67    StatusLine=20
   68    Top = 20: Left = 450
   69    MaxBobRight=3*.8*FRE(0)/4 : MaxBobBottom=.8*FRE(0)/
4
   70    RETURN
   71
   72  InitFile:
   73    CLS
   74    FileName$=""
   75    Flags=SAVEBACK+OVERLAY+fVSprite
   76    IF fVSprite = 1 THEN BobRight=15 ELSE BobRight=31
   77    BobBottom=31
   78    CurrentX=BobRight:CurrentY=BobBottom
   79    maxColor=2^Depth - 1
   80    PlanePick=maxColor
   81    Change=0
   82    RETURN
   83
   84  InitMenu:
   85    MENU 1,0,1,"File"
   86    MENU 1,1,1,"New"
   87    MENU 1,2,1,"Open ..."
   88    MENU 1,3,1,"Save"
   89    MENU 1,5,1,"Quit"
   90    MENU 1,4,1,"Save as ..."
   91    MENU 2,0,1,"Tools"
   92    MENU 3,0,1,"Test"
   93    MENU 3,1,1,"Object On"
   94    MENU 3,2,1,"Object Off"
   95    MENU 4,0,1,""
   96
   97    ToolName$(1)="Pen"
   98    ToolName$(2)="BigPen"
   99    ToolName$(3)="Line"
  100    ToolName$(4)="Circle"
  101    ToolName$(5)="Square"
  102    ToolName$(6)="Erase"
  103    ToolName$(7)="Fill"
  104    FOR i=1 TO MaxTool
  105      MENU 2,i,1,ToolName$(i)
  106    NEXT i
  107
```

```
  108    RETURN
  109
  110  CheckMenu:
  111    MenuId=MENU(0)
  112    MenuItem=MENU(1)
  113    ON MenuId GOTO FileMenu,ToolsMenu,Test
  114
  115  CheckMouse:
  116    GetCurrentXY
  117    IF CurrentY>MaxY+10 THEN CheckColor
  118    IF NOT fEnlarge THEN
  119      IF CurrentY>BobBottom+10 OR CurrentX>BobRight+10
THEN RETURN
  120      IF CurrentY>=BobBottom AND CurrentX>=BobRight THE
N ChangeSizePicture
  121      IF (CurrentY>BobBottom OR CurrentX>BobRight) THEN
 RETURN
  122    ELSE
  123      IF CurrentX>BobRight*Offset OR CurrentY>BobBottom
*Offset THEN RETURN
  124    END IF
  125    StartY=CurrentY
  126    StartX=CurrentX
  127    Change=-1
  128    ON ToolMode GOSUB Pen,BigPen,DrawLine,DrawCircle,Dr
awRectangle,ErasePicture,PaintPicture
  129    RETURN
  130
  131  DrawLine:
  132    WHILE MOUSE(0)<>0
  133      GetCurrentXY
  134      IF InsideBob THEN
  135        InvertVideo
  136        LINE (StartX-1,StartY)-(CurrentX-1,CurrentY)
  137        LINE (StartX-1,StartY)-(CurrentX-1,CurrentY)
  138        NormalVideo
  139      END IF
  140    WEND
  141    LINE (StartX-1,StartY)-(CurrentX-1,CurrentY),Curren
tColor
  142    RETURN
  143
  144  SUB ChangeToolsMode (Mode) STATIC
  145  SHARED MaxTool
  146    FOR i=2 TO MaxTool
  147      MENU 2,i,Mode
  148    NEXT
  149  END SUB
  150
  151  Pen:
  152    GetCurrentXY
  153    IF InsideBob THEN PSET (CurrentX-1,CurrentY),Curren
tColor
  154    WHILE MOUSE(0)<>0
  155      GetCurrentXY
  156      IF NOT InsideBob THEN GOTO FarbenNeu
  157      LINE -(CurrentX-1,CurrentY),CurrentColor
  158    WEND
  159    RETURN
  160
  161  FarbenNeu:
  162    WHILE MOUSE(0)><0
  163      GetCurrentXY
  164      LOCATE 20,10:PRINT CurrentX" "CurrentY
  165      WEND
  166    RETURN
  167
  168  BigPen:
  169    GetCurrentXY
  170    IF InsideBob THEN LINE (CurrentX-2,CurrentY-1)-(Cur
rentX,CurrentY+1),CurrentColor,bf
  171    FatX=CurrentX:FatY=CurrentY
  172    WHILE MOUSE(0)<>0
  173    GetCurrentXY
  174    IF NOT InsideBob THEN GOTO FarbenNeu
  175    LINE (FatX-2,FatY-1)-(CurrentX-2,CurrentY-1),Curren
tColor
  176    LINE (FatX-1,FatY-1)-(CurrentX-1,CurrentY-1),Curren
tColor
  177    LINE (FatX,FatY-1)-(CurrentX,CurrentY-1),CurrentCol
or
  178    LINE (FatX-2,FatY)-(CurrentX-2,CurrentY),CurrentCol
or
  179    LINE (FatX-1,FatY)-(CurrentX-1,CurrentY),CurrentCol
or
  180    LINE (FatX,FatY)-(CurrentX,CurrentY),CurrentColor
  181    LINE (FatX-2,FatY+1)-(CurrentX-2,CurrentY+1),Curren
tColor
  182    LINE (FatX-1,FatY+1)-(CurrentX-1,CurrentY+1),Curren
tColor
  183    LINE (FatX,FatY+1)-(CurrentX,CurrentY+1),CurrentCol
or
  184    FatX=CurrentX:FatY=CurrentY
  185    WEND
  186    RETURN
  187
  188  DrawCircle:
  189    GOSUB TrackRect1
  190    CenterX=StartX-1
  191    CenterY=StartY
  192    RadiusX=ABS(CurrentX-StartX)
  193    RadiusY=ABS(CurrentY-StartY)
  194    IF RadiusX=0 OR RadiusY=0 THEN RETURN
  195    Aspect!=ABS(RadiusY/RadiusX)
  196    IF RadiusX < RadiusY THEN RadiusX=RadiusY
  197    CIRCLE (CenterX,CenterY),RadiusX,CurrentColor,,,Asp
ect!
  198    RETURN
  199
  200  DrawRectangle:
  201    GOSUB TrackRect
  202    LINE (DrawRect(1),DrawRect(0))-(DrawRect(3),DrawRec
t(2)),CurrentColor,b
  203    RETURN
  204
  205  ErasePicture:
  206    WHILE MOUSE(0)<>0
  207      GetCurrentXY
  208      IF CurrentX-2<0 OR CurrentY-1<0 THEN InsideBob=0
  209      IF InsideBob THEN
  210        LINE (CurrentX-2,CurrentY-1)-(CurrentX,CurrentY
+1),1,bf
  211        LINE (CurrentX-2,CurrentY-1)-(CurrentX,CurrentY
+1),0,bf
  212      END IF
  213    WEND
  214    DrawBoundary
  215    RETURN
  216
  217  PaintPicture:
  218    IF InsideBob THEN
  219      LINE(0,BobBottom+1)-(BobRight+1,BobBottom+1),Cur
rentColor
  220      LINE(BobRight+1,0)-(BobRight+1,BobBottom+1),Curr
entColor
  221      PAINT (CurrentX, CurrentY),CurrentColor
  222      DrawBoundary
  223    END IF
  224    ToolMode=1:GOSUB PrintToolStatus
  225    RETURN
  226
  227  TrackRect:
  228    WHILE MOUSE(0)<>0
  229      GetCurrentXY
  230      IF InsideBob THEN
  231        DrawRect(0)=StartY
  232        DrawRect(1)=StartX-1
  233        DrawRect(2)=CurrentY
  234        DrawRect(3)=CurrentX-1
  235        InvertVideo
  236        FrameRect DrawRect()
  237        FrameRect DrawRect()
  238        NormalVideo
  239      END IF
  240    WEND
  241    IF CurrentY<StartY THEN DrawRect(0)=CurrentY: DrawR
ect(2)=StartY
  242    IF CurrentX<StartX THEN DrawRect(1)=CurrentX-1: Dra
wRect(3)=StartX-1
  243    RETURN
  244
  245  TrackRect1:
  246    WHILE MOUSE(0)<>0
  247      GetCurrentXY
  248      IF InsideBob THEN
  249        DrawRect(0)=StartY-ABS(CurrentY-StartY)
```

```
  250         DrawRect(1)=StartX-1-ABS(CurrentX-StartX)
  251         DrawRect(2)=StartY+ABS(CurrentY-StartY)
  252         DrawRect(3)=StartX-1+ABS(CurrentX-StartX)
  253         InvertVideo
  254         FrameRect DrawRect()
  255         FrameRect DrawRect()
  256         NormalVideo
  257       END IF
  258     WEND
  259     IF CurrentY<StartY THEN DrawRect(0)=CurrentY: DrawR
ect(2)=StartY
  260     IF CurrentX<StartX THEN DrawRect(1)=CurrentX-1: Dra
wRect(3)=StartX-1
  261     RETURN
  262
  263   ChangeSizePicture:
  264     MaxMem = .8 * FRE(0)
  265     COLOR 0
  266     DrawBoundary
  267     COLOR 1
  268     InvertVideo
  269     WHILE MOUSE(0)<>0
  270       GetCurrentXY
  271       IF (CurrentY < MaxY) AND (CurrentY > 0) THEN
  272        IF (CurrentX <= MaxX) AND (CurrentX >= 10) THEN
  273         IF MaxMem > (Depth * CurrentX * CurrentY /8) TH
EN
  274           IF fVSprite = 1 THEN BobRight = 15:CurrentX=1
5:ELSE BobRight=CurrentX
  275           BobBottom=CurrentY
  276           DrawBoundary
  277           DrawBoundary
  278         END IF
  279        END IF
  280       END IF
  281     WEND
  282     NormalVideo
  283     GOSUB GetPicture
  284     GOSUB RedrawPicture
  285     RETURN
  286
  287   ToolsMenu:
  288     ToolMode=MenuItem
  289     GOSUB PrintToolStatus
  290     RETURN
  291
  292   FileMenu:
  293     ON MenuItem GOSUB NewFile,OpenFile,SaveFile,SaveFil
eAs,Quit
  294     RETURN
  295
  296   Test:
  297     ON MenuItem GOSUB ObjOn,ObjOff
  298     IF MenuItem = 2 THEN GOSUB InitFile : GOTO StartOve
r
  299     RETURN
  300
  301   NewFile:
  302     GOSUB CheckSave
  303     IF CancelCommand THEN RETURN
  304     CLS
  305     GOSUB InitFile
  306     GOTO StartOver
  307
  308   ObjOn:
  309     CLS
  310     INPUT "Dateinamen eingeben > ",FileName$
  311     IF FileName$="" THEN NewFile
  312     OPEN FileName$ FOR INPUT AS 1
  313     OBJECT.SHAPE 1,INPUT$(LOF(1),1)
  314     CLOSE 1
  315     GOSUB LoadCol
  316     OBJECT.X 1,100
  317     OBJECT.Y 1,50
  318     OBJECT.VX 1,8
  319     OBJECT.VY 1,2
  320     OBJECT.ON
  321     OBJECT.START
  322     RETURN
  323
  324   ObjOff:
  325     OBJECT.OFF 1
  326     CLS
  327     RETURN
  328
  329   OpenFile:
  330     GOSUB CheckSave
  331     IF CancelCommand THEN RETURN
  332     CLS
  333     INPUT "Dateinamen eingeben > ",FileName$
  334     IF FileName$="" THEN NewFile
  335     OPEN FileName$ FOR INPUT AS 1
  336     ColorSet=CVL(INPUT$(4,1))
  337     DataSet=CVL(INPUT$(4,1))
  338     Depth=CVL(INPUT$(4,1))
  339     BobRight=CVL(INPUT$(4,1)) - 1
  340     BobBottom=CVL(INPUT$(4,1)) - 1
  341     REM --- UNDONE if ColorSet<>0 or DataSet<>0, read i
mage.editor format file
  342     Flags=CVI(INPUT$(2,1))
  343     IF Flags AND 1 THEN fVSprite = 1 ELSE fVSprite = 0
  344     IF PlanePick <> CVI(INPUT$(2,1)) THEN
  345       PRINT "Fehler: Datei hat nicht Auflösung/Bit-Plan
e-Anzahl dieses SCREENs"
  346     ELSE
  347       PlaneOnOff=CVI(INPUT$(2,1))
  348       ArraySize&=FNArraySize&
  349       DIM BobArray(ArraySize&)
  350       BobArray(0)=BobRight + 1
  351       BobArray(1)=BobBottom + 1
  352       BobArray(2)=Depth
  353       FOR i=3 TO ArraySize&-1
  354         BobArray(i)=CVI(INPUT$(2,1))
  355       NEXT i
  356       CLS
  357       CurrentX=BobRight: CurrentY=BobBottom
  358       GOSUB RedrawPicture
  359     END IF
  360     CLOSE #1
  361     GOSUB LoadCol
  362     Change=0
  363     GOTO StartOver
  364
  365   LoadCol:
  366     ColName$=FileName$+".Col"
  367     OPEN ColName$ FOR INPUT AS 1
  368     FOR i=0 TO 31
  369       FOR j=0 TO 2
  370       INPUT#1,x$:Col(i,j)=VAL(x$)
  371       NEXT
  372       PALETTE i,Col(i,0)/16,Col(i,1)/16,Col(i,2)/16
  373     NEXT
  374     CLOSE#1
  375     RETURN
  376
  377   SaveFileAs:
  378     FileName$=""
  379
  380   SaveFile:
  381     GOSUB GetPicture
  382     IF FileName$="" THEN CLS: INPUT "Dateinamen eingebe
n > ",FileName$
  383     IF FileName$<>"" THEN
  384       OPEN FileName$ FOR OUTPUT AS 1
  385       PRINT #1, MKL$(0); 'ColorSet
  386       PRINT #1, MKL$(0); 'DataSet
  387       PRINT #1, MKI$(0);MKI$(BobArray(2)); 'Tiefe
  388       PRINT #1, MKI$(0);MKI$(BobArray(0)); 'Breite
  389       PRINT #1, MKI$(0);MKI$(BobArray(1)); 'Höhe
  390       PRINT #1, MKI$(Flags);
  391       PRINT #1, MKI$(PlanePick);  'planePick
  392       PRINT #1, MKI$(0);  'planeOnOff
  393       FOR i=3 TO ArraySize&-1
  394         PRINT #1, MKI$(BobArray(i));
  395       NEXT i
  396       CLOSE#1
  397     END IF
  398
  399     ColName$=FileName$+".Col"
  400     OPEN ColName$ FOR OUTPUT AS 1
  401     FOR i=0 TO 31
  402       FOR j=0 TO 2
  403       PRINT#1,STR$(Col(i,j))
  404       NEXT
  405     NEXT
  406     CLOSE#1
```

```
407
408     GOSUB RedrawPicture
409     Change=0
410     RETURN
411
412   Quit:
413     Cancel=0
414     GOSUB CheckSave
415     IF CancelCommand THEN RETURN
416     Unfinished=0
417     RETURN
418
419   GetPicture:
420     ArraySize&=FNArraySize&
421     DIM BobArray(ArraySize&)
422     GET (0,0)-(BobRight,BobBottom),BobArray
423     RETURN
424
425   RedrawPicture:
426     CLS
427     PUT (0,0),BobArray,PSET
428     ERASE BobArray
429     DrawBoundary
430     GOSUB PrintStatus
431     RETURN
432
433   PrintStatus:

434     PrintCurrentXY
435     GOSUB PrintToolStatus
436     GOSUB PrintColorBar
437     RETURN
438
439   PrintToolStatus:
440     LOCATE StatusLine,24: PRINT SPACE$(10);
441     LOCATE StatusLine,24: PRINT ToolName$(ToolMode);
442     RETURN
443
444   PrintColorBar:
445     COLOR CurrentColor
446     LOCATE 19,1: PRINT "Farbe:";
447     ColorBar = WINDOW(5)-10
448     COLOR 1
449     x=70
450     FOR i=0 TO maxColor
451       LINE (x,ColorBar)-(x+7,y+ColorBar+10),i,bf
452       LINE (x,ColorBar)-(x+7,y+ColorBar+10),1,b
453       x=x+7
454     NEXT i
455     GOSUB ShowPalette
456     RETURN
457
458   CheckColor:
459     IF CurrentY<ColorBar THEN RETURN
460     IF CurrentY>ColorBar+20 THEN GOTO ChangePalette
461     IF CurrentX<70 THEN RETURN
462     i=INT((CurrentX-70)/7)
463     IF i>maxColor THEN RETURN
464     CurrentColor=i
465     GOSUB PrintColorBar
466     RETURN
467
468   ShowPalette:
469     LOCATE StatusLine+2,1:PRINT "+++++++++++++++++  R"Co
l(CurrentColor,0)
470     LOCATE StatusLine+2,Col(CurrentColor,0)+1:PRINT "*"

471     LOCATE StatusLine+3,1:PRINT "+++++++++++++++++  G"Co
l(CurrentColor,1)
472     LOCATE StatusLine+3,Col(CurrentColor,1)+1:PRINT "*"
473     LOCATE StatusLine+4,1:PRINT "+++++++++++++++++  B"Co
l(CurrentColor,2)
474     LOCATE StatusLine+4,Col(CurrentColor,2)+1:PRINT "*"
475     RETURN
476
477   ChangePalette:
478     CC=CurrentColor:CX=INT(CurrentX/8)
479     IF CX>15 THEN CX = 15
480     CY=INT((CurrentY-168)/8):IF CY>2 THEN RETURN
481     Col(CC,CY)=CX
482     PALETTE CC,Col(CC,0)/16,Col(CC,1)/16,Col(CC,2)/16
483     GOSUB ShowPalette
484     RETURN
485
486   CheckSave:
487     IF fEnlarge THEN GOSUB Shrink
488     CancelCommand=0
489     IF Change THEN
490       BEEP
491       GOSUB GetPicture
492       CLS
493       PRINT "Aktuelle Daten sind nicht abgespeichert."
494       PRINT "Wollen Sie sie speichern?"
495       PRINT " Y-Taste betätigen zum Speichern,"
496       PRINT " N-Taste speichert nicht (Daten weg),"
497       PRINT " C-Taste bricht Befehl ab."
498       Response=0
499       WHILE Response=0
500         a$=INKEY$
501         IF a$<>"" THEN
502           a$=UCASE$(a$)
503           IF a$="Y" THEN Response=1
504           IF a$="N" THEN Response=2
505           IF a$="C" THEN Response=3
506           IF Response=0 THEN BEEP
507         END IF
508       WEND
509       GOSUB RedrawPicture
510       IF Response=1 THEN GOSUB SaveFileAs
511       IF Response=3 THEN CancelCommand=-1
512     END IF
513     RETURN
514
515   SUB GetCurrentXY STATIC
516     SHARED CurrentX,CurrentY,InsideBob,BobRight,BobBott
om
517     dummy=MOUSE(0)
518     CurrentX=MOUSE(1)
519     CurrentY=MOUSE(2)
520     InsideBob=-1
521     IF CurrentX>BobRight OR CurrentY>BobBottom THEN Ins
ideBob=0
522     IF CurrentX<0 OR CurrentY<0 THEN InsideBob=0
523     END SUB
524
525   SUB PrintCurrentXY STATIC
526     SHARED StatusLine,CurrentX,CurrentY
527     LOCATE StatusLine,1: PRINT "BOB-Size X:";CurrentX;
528     LOCATE StatusLine,17: PRINT "Y:";CurrentY;
529   END SUB
530
531   SUB DrawBoundary STATIC
532   SHARED BobRight,BobBottom
533     x=BobRight+10
534     y=BobBottom+10
535     LINE (0,y)-(x,y)
536     LINE (x,y)-(x,0)
537     LINE (0,BobBottom+1)-(x,BobBottom+1)
538     LINE (BobRight+1,y)-(BobRight+1,0)
539   END SUB
540
541   SUB InvertVideo STATIC
542      CALL SetDrMd& (WINDOW(8),3)
543   END SUB
544
545   SUB NormalVideo STATIC
546      CALL SetDrMd& (WINDOW(8),1)
547   END SUB
548
549   SUB FrameRect(rect()) STATIC
550     LINE (rect(1),rect(0))-(rect(3),rect(0))
551     LINE (rect(3),rect(0))-(rect(3),rect(2))
552     LINE (rect(3),rect(2))-(rect(1),rect(2))
553     LINE (rect(1),rect(2))-(rect(1),rect(0))
554   END SUB
555
556   IgnoreBreak:
557     RETURN
```

# Neue Kickstart-Routinen

**Liste der neuen KS 1.2 Routinen**

*Nach den neuen Intuition-Routinen im letzten Heft kommen wir diesmal zu den neuen Grafik-Routinen.*

AndRegionRegion

Und-Verknüpfung von zwei Regions

```
Erg = AndRegionRegion (Region1,Region2)
d0                    a0      a1

BOOL Erg;
struct Region *Region1,*Region2;
```

Library-Offset:
-624(GfxBase)

Funktion:
Diese Funktion bewirkt eine 2-dimensionale Und-Verknüpfung von zwei Regions. Das Ergebnis der Verknüpfung wird in Region2 abgelegt, Region1 bleibt unverändert.

Eingaben:
Region1, Region2: Zeiger auf die zwei zu verknüpfenden Region-Strukturen.

Ergebnis:
Erg TRUE bei erfolgreicher Durchführung, ansonsten FALSE.

Bugs:

Querverweis:
AndRectRegion()

ClearRectRegion

Löscht eine rechteckigen Bereich einer Region

```
Erg = ClearRectRegion (Region, RechtE)
d0                     a0      a1

BOOL Erg;
struct Region *Region;
struct Rectangle *RechtE;
```

Library-Offset:
-522(GfxBase)

Funktion:
Löscht einen rechteckigen Bereich der angegebenen Region.

Eingaben:
Region Zeiger auf die Region_Struktur der zu verändernden Region.
RechtE Zeiger auf eine Rectangle-Struktur.

Ergebnis:
Erg TRUE bei efolgreicher Durchführung, sonst FALSE.

Bugs:

Querverweis:
AndRectRegion(), graphics/regions.h

OrRegionRegion

Oder-Verknüpfung von zwei Regions

```
Erg = OrRegionRegion (Region1, Region2)
d0                    a0       a1

BOOL Erg;
struct Region *Region1,*Region2;
```

Library-Offset:
-612(GfxBase)

Funktion:
Diese Funktion bewirkt eine Oder-Verknüpfung der beiden Regions. Die daraus resultierende Region wird in Region2 abgespeichert, Region1 bleibt unverändert.

Eingaben:
Region1, Region2 Zeiger auf die Region-Strukturen der beiden Regions.

Ergebnis:
Erg TRUE bei efolgreicher Durchführung, sonst FALSE.

Bugs:

Querverweis:

XorRegionRegion

Exklusiv-Oder Verknüpfung von zwei Regions

```
Erg = XorRegionRegion (Region1, Region2)
d0                    a0       a1

BOOL Erg;
struct Region *Region1,*Region2;
```

Library-Offset:
-618(GfxBase)

Funktion:
Exklusiv-Oder Verknüpfung von zwei Regions. Das Ergebnis der Verknüpfung wird in Region2 abgelegt, Region1 bleibt unverändert.

Eingaben:
Region1,Region2 Zeiger auf die Region-Strukturen der beiden Regions.

Ergebnis:
Erg TRUE bei efolgreicher Durchführung, sonst FALSE.

Bugs:

Querverweis:

AreaEllipse

Fügt eine Ellipse in die AreaInfo Liste ein

# Neue Kickstart-Routinen

```
     Erg = AreaEllipse (rp, cx,   cy,   a,    b)
     d0                 a1  d0:16 d1:16 d2:16 d3:16

     long Erg;
     struct RastPort *rp;
     SHORT cx,cy,a,b;

Library-Offset:
     -186(GfxBase)

Funktion:
     Fügt eine Ellipse in die AreaInfo Liste zum
     späteren Zeichnen und Füllen ein.

Eingaben:
     rp        Zeiger auf den Rastport, in den
               gezeichnet werden soll.
     cx,cy     Mittelpunkt der Ellipse relativ zum
               RastPort.
     a         Horizontaler Radius der Ellipse
               (a>0!!!)
     b         Vertikaler Radius der Ellipse (b>0!!!)

     Achtung! AreaEllipse führt kein Clipping durch !

Ergebnis:
     Erg       0 wenn kein Fehler auftrat, -1 wenn
               kein Platz in der Liste mehr ist.

Bugs:

Querverweis:
     AreaMove(),AreaDraw(), AreaCircle(), InitArea(),
     graphics/ rastport.h, graphics/gfxmacros.h
```

AreaCircle

Fügt einen Kreis in die AreaInfo Liste ein

```
     Erg = AreaCircle (rp, cx, cy, r)
     d0                a1  d0  d1  d2

     long Erg;
     struct RastPort *rp;
     SHORT cx,cy,r;

Library-Offset:
     AreaCircle ist ein Macro, das AreaEllipse-
     (rp,cx,cy,r,r) aufruft.

Funktion:
     Fügt einen Kreis in die AreaInfo Liste ein.

Eingaben:
     rp        Zeiger auf den RastPort, in den
               gezeichnet werden soll.
     cx,cy     Mittelpunkt des Kreises relativ zum
               RastPort.
     r         Radius des Kreises.

Ergebnis:
     Erg       0 wenn kein Fehler auftrat, -1 wenn
               kein Platz in der Liste mehr ist.

Bugs:

Querverweis:
     AreaMove(),AreaDraw(), AreaCircle(), InitArea(),
     graphics/ rastport.h, graphics/gfxmacros.h
```

DrawEllipse

Zeichnet eine Ellipse

```
     DrawEllipse (rp, cx, cy, a, b)
                  a0  d0  d1  d2 d3

     struct RastPort *rp;
     SHORT cx,cy,a,b;

Library-Offset:
     -180(GfxBase)

Funktion:
     Zeichnet eine Ellipse um den Punkt (cx,cy) mit
     den Radien a und b.

Eingaben:

     rp        Zeiger auf den RastPort, in den
               gezeichnet werden soll.
     cx,cy     Mittelpunkt der Ellipse.
     a         Horizontaler Radius der Ellipse
               (a>0!!!).
     b         Vertikaler Radius der Ellipse
               (b>0!!!).

     Achtung! DrawEllipse führt kein Clipping durch!

Ergebnis:

Bugs:

Querverweis:
     DrawCircle(), graphics/rastport.h
```

DrawCircle

Zeichnet einen Kreis

```
     DrawCircle (rp, cx, cy, r)
                 a0  d0  d1  d2

     struct RastPort *rp;
     SHORT cx,cy,r;

Library-Offset:
     Diese Funktion ist ein Macro, das DrawEllipse
     (rp,cx,cy,r,r) aufruft.

Funktion:
     Zeichnet einen Kreis mit dem Mittelpunkt (cx,cy)
     und dem Radius r.

Eingaben:
     rp        Zeiger auf den RastPort, in den
               gezeichnet werden soll.
     cx,cy     Mittelpunkt des Kreises.
     r         Radius des Kreises.

Bugs:

Querverweis:
     DrawEllipse(), graphics/rastport.h
```

AttemptLockLayerRom

```
     erg = AttemptLockLayerRom(layer)
     d0                        a5

     BOOL erg;
     struct Layer *layer;
```

```
Library-Offset:
     -654(GfxBase)

Funktion:
     überprüft den Lock-Status der Layer. Falls sie
     schon geLOCKt ist, wird False zurückgegeben,
     ansonsten wird sie geLOCKt und TRUE zurück-
     gegeben. Dabei werden keine anderen Register
     zerstört.

Eingaben:
     layer     Zeiger auf eine Layer-Struktur

Ergebnis:
     erg       True, falls geLOCKt werden konnte,
               ansonsten FALSE

Bugs:

Querverweis:
     LockLayerRom(), UnlockLayerRom()
```

InstallClipRegion

Installiert eine Region in einer Layer

```
     oldClip = InstallClipRegion (layer,region)
     d0                          a0    a1

     struct Region *oldClip;
     struct Layer *layer;
     struct Region *region;

Funktion:
     Installiert eine Region in einer Layer. alle
     folgenden Grafikzugriffe werden dann bezüglich
     dieser Region geclipped.

Eingaben:
     layer     Zeiger auf die Layer, in der die
               Region eingebunden werden soll.
     region    Zeiger auf die Region, die eingebunden
               werden soll.

Ergebnis:
     oldClip   Bei erfolgreicher Durchführung steht
               hier der Zeiger auf die alte Region
               drin, wenn keine Region eingebunden
               war, wird 0 zurückgegeben.

Bugs:

Querverweis:
```

BlitBitMapRastPort

Kopieren BitMap -> RastPort

```
     BlitBitMapRastPort       (srcbm,sx,sy,destrp,dx,
                                   dy,cx,cy,minterm)
                         a0    d0 d1 a1     d2
                                   d3 d4 d5 d6

     struct BitMap *srcbm;
     SHORT sx,sy;
     struct Rastport *destrp;
     SHORT dx,dy,cx,cy;
     UBYTE minterm;

Library-Offset:
     -606(GfxBase)

Funktion:
     Kopiert einen Block aus einer BitMap in einen
     RastPort, Modus= minterm.

Eingaben:
     srcbm     Zeiger auf die Source-Bitmap
     sx,sy     linke obere Ecke der Source des Blocks
     destrp    Zeiger auf den Destination-Rastport
     dx,dy     linke obere Ecke der Destination des
               Blocks
     cx,cy     Breite/Höhe des Blocks
     minterm   Blitter-Modus

Ergebnis:
     TRUE

Bugs:

Querverweis:
     BlitMaskBitMapRastPort(), graphics/gfx.h,
     graphics/rastport.h
```

BlitMaskBitMapRastPort

BlitBitMapRastPort mit Maske für Source

```
     BlitMaskBitMapRastPort
          (srcbm,sx,sy,destrp,dx,dy,cx,cy,minterm,
                                              bltmask)
          a0     d0 d1 a1     d2 d3 d4 d5 d6 a2

     struct BitMap *srcbm;
     SHORT sx,sy;
     struct Rastport *destrp;
     SHORT dx,dy,cx,cy;
     UBYTE minterm;
     APTR bltmask;       !CHIP-MEMORY!

Funktion:
     Kopiert einen Block aus einer BitMap in einen
     RastPort mit Modus=minterm und der Maske
     bltmask.

Eingaben:
     srcbm     Zeiger auf die Source-Bitmap
     sx,sy     linke obere Ecke der Source des Blocks
     destrp    Zeiger auf den Destination-Rastport
     dx,dy     linke obere Ecke der Destination des
               Blocks
     cx,cy     Breite/Höhe des Blocks
     minterm   Blitter-Modus
     bltmask   Zeiger auf eine einzelne Bitplane, die
               die selbe Größe wie der zu kopierende
               Block haben muß.

Ergebnis:

Bugs:

Querverweis:
     BlitBitMapRastPort(), graphics/gfx.h, graphics
     /rastport.h
```

SetRGB4CM

Setzen eines Farb-Register für eine Colormap

```
     SetRGB4CM (cm,n ,r ,  g ,  b)
               a0 d0 d1:4 d2:4 d3:4

     struct ColorMap *cm;
     short n;
     UBYTE r,g,b;
```

```
Library-Offset:
     -630(GfxBase)

Funktion:
     Schreibt das Triple (r,g,b)  an der  Stelle n in
     eine ColorMap Struktur.

Eingaben:
     cm     Zeiger auf die ColorMap Struktur
     n      Nummer der Farbe
     r,g,b  Farbanteile

Ergebnis:

Bugs:

Querverweis:
     GetColorMap(),  GetRGB4(),  SetRGB(),  graphics/
     view.h
```

# Nimbi

Wer in der Computerszene hat nicht schon einmal das Spiel 'NIM' gespielt! Schnell fand man Gefallen daran und genauso schnell verlor das Spiel jeglichen Reiz, nachdem man nach wenigen Spielen die optimale Spielstrategie erkannt hatte. Wenig bekannt ist eine zweidimensionale Variante - das Spiel 'NIMBI', das von den Spielregeln genauso einfach ist, für das aber eine optimale Strategie nur schwer zu finden ist.

Wie 'NIM' ist auch 'NIMBI' ein Zwei-Personen-Spiel. Gespielt wird auf einem Spielfeld, das aus 12 runden Feldern besteht, die wie im Bild, durch Linien verbunden sind.

Zu Beginn steht auf jedem Feld genau ein Stein. Aufgabe der Spieler ist es nun, abwechselnd einen oder mehrere Steine zu entfernen, bis kein Stein mehr auf einem Feld vorhanden ist. Mehrere Steine können nur dann entfernt werden, wenn Sie auf einer geraden Linie liegen und eine ununterbrochene Kette bilden, so daß kein leeres Feld dazwischen liegt. Verlierer ist, wer den letzten Stein entfernen muß.

Das BASIC-Programm ermöglicht das Spiel von zwei Personen gegeneinander. Das Spiel gegen den Computer ist hier nicht realisiert, da das Programm in diesem Fall zu umfangreich geworden wäre.

Nach dem Start des Programms und dem Erscheinen des Titels, stehen zwei Menues zur Verfügung. Das erste regelt im wesentlichen den Spielaufbau:

Das Erstellen des Spielfeldes, die Eingabe der Spieler und der Spielstart sind - soweit erforderlich - in einer festen Reihenfolge zu durchlaufen.

Zu Beginn eines Spieles kann man sich noch einmal die Spielregeln ansehen. Auch kann ein laufendes Spiel jederzeit mit 'Neubeginn' abgebrochen und neu gestartet werden.

Das zweite Menue dient dem Spielablauf:

Steine werden dadurch entfernt, daß sie zunächst mit der Maus markiert werden - sie erscheinen dann blau - und anschließend im zweiten Menue der Menuepunkt 'Entfernen' gewählt wird. Mit 'Undo' kann eine vorgenommene Markierung rückgängig gemacht werden.

Die Reihenfolge der Spieler wird durch ein rotes Rechteck sowie einen Ton angezeigt. Beim Spielen ist unbedingt darauf zu achten, daß der Spieler, der an der Reihe ist, eine Markierung von Feldern mit der Maus erst dann vornimmt, wenn für ihn das rote Rechteck angezeigt wird.

```
  1  'Spiel NIMBI
  2  'Hubert Wagner
  3  'Bitte 80-Zeichen aktivieren
  4
  5  SCREEN 2,640,400,3,2
  6  WINDOW 2,"NIMBI",(0,0) - (620,180),0,2
  7  PALETTE 5,.93,.2,0!  : PALETTE 6,0!,.93,.87
  8  DIM mark%(4,4), xpos%(4,4), ypos%(4,4)
  9  vonneuem%=1
 10  GOSUB Wertefestlegung
 11  GOSUB TITEL
 12  GOSUB Menuleiste
 13
 14  Hauptschleife:
 15
 16  i% = 1
 17  IF vonneuem% = 0 THEN MENU 1,3,1 : MENU 1,4,0
 18
 19  WHILE i% <> 0
 20    ON MENU GOSUB InfoAnzeige
 21    MENU ON
 22    SLEEP
 23  WEND
 24
 25  WHILE Abbruch% < 2
 26
 27    Abbruch% = 0
 28    WHILE Abbruch% = 0
 29
 30    ON MENU GOSUB Menuauswahl
 31    MENU ON
 32
 33    WEND
 34    GOSUB Wahl1
 35
 36  WEND
 37
 38  Spielstart:
 39
 40  vonneuem% = 1
 41  GOSUB Nullsetzen
 42  GOSUB Spielereintrag
 43  GOSUB Markensetzen
 44
 45  anzmarken%=12 : reihenf%=1
 46  WHILE anzmarken% > 0
 47    IF reihenf%=1 THEN GOSUB ErsterSp ELSE GOSUB Zweite
rSP
 48
 49    Markieren:
 50
 51      test%=0 : stzahl% = 0
 52      WHILE test% < 2
 53
 54        ON MENU GOSUB Checkmenu
 55        ON MOUSE GOSUB Steinmark
 56        MENU ON
 57        MOUSE ON
 58
 59      WEND
 60
 61  WEND
 62
 63  GOSUB Gewinner
 64
 65  END
 66
 67
 68  TITEL:
 69    PALETTE 1,.73,1!,0!
 70    COLOR 1
 71    LINE (40,20) - (50,120),1,BF
 72    LINE (90,20) - (100,120),1,BF
 73    AREA (40,20) : AREA (50,20) : AREA (100,120) : AREA
(90,120)
 74    AREAFILL
 75    LINE (150,20) - (160,120),1,BF
 76    LINE (210,20) - (220,120),1,BF
 77    LINE (280,20) - (290,120),1,BF
 78    AREA (210,20) : AREA (220,20) : AREA (250,90) : ARE
```

```
A (280,20) : AREA (290,20) : AREA (250,110)
   79    AREAFILL
   80    LINE (340,20) - (350,120),1,BF
   81    LINE (350,20) - (400,24),1,BF
   82    LINE (350,68) - (400,72),1,BF
   83    LINE (350,116) - (400,120),1,BF
   84    AREA (400,20) : AREA (408,27) : AREA (414,33) : ARE
A (418,39) : AREA (420,45)
   85    AREA (418,51) : AREA (414,57) : AREA (408,63) : ARE
A (400,70) : AREA (390,70)
   86    AREA (398,63) : AREA (404,57) : AREA (408,51) : ARE
A (410,45) : AREA (408,39)
   87    AREA (404,33) : AREA (398,27) : AREA (390,20)
   88    AREAFILL
   89    AREA (400,70) : AREA (408,77) : AREA (414,83) : ARE
A (418,89) : AREA (420,95)
   90    AREA (418,101) : AREA (414,107) : AREA (408,113) :
AREA (400,120) : AREA (390,120)
   91    AREA (398,113) : AREA (404,107) : AREA (408,101) :
AREA (410,95) : AREA (408,89)
   92    AREA (404,83) : AREA (398,77) : AREA (390,70)
   93    AREAFILL
   94    LINE (470,20) - (480,120),1,BF
   95
   96  RETURN
   97
   98  Spielfeld:
   99
  100    IF vonneuem% = 0 THEN RETURN
  101    CLS
  102    CIRCLE (170,150),25,3 : CIRCLE (320,150),25,3 :CIRC
LE (470,150),25,3
  103    PAINT (170,150),3     : PAINT (320,150),3     :PAIN
T (470,150),3
  104
  105    CIRCLE (95,110),25,3 : CIRCLE (245,110),25,3 :CIRCL
E (395,110),25,3
  106    CIRCLE (545,110),25,3
  107    PAINT (95,110),3      : PAINT (245,110),3     :PAINT
 (395,110),3
  108    PAINT (545,110),3
  109
  110    CIRCLE (170,70),25,3 : CIRCLE (320,70),25,3 :CIRCLE
 (470,70),25,3
  111    PAINT (170,70),3      : PAINT (320,70),3      :PAINT
(470,70),3
  112
  113    CIRCLE (245,30),25,3 :CIRCLE (395,30),25,3
  114    PAINT (245,30),3      :PAINT (395,30),3
  115
  116    LINE (170,150) - (470,150),3
  117    LINE (95,110) - (545,110),3
  118    LINE (170,70) - (470,70),3
  119    LINE (245,30) - (395,30),3
  120
  121    LINE (95,110) - (245,30),3
  122    LINE (170,150) - (395,30),3
  123    LINE (320,150) - (470,70),3
  124    LINE (470,150) - (545,110),3
  125
  126    LINE (170,150) - (95,110),3
  127    LINE (320,150) - (170,70),3
  128    LINE (470,150) - (245,30),3
  129    LINE (545,110) - (395,30),3
  130
  131    MENU 1,5,0 : MENU 1,1,0 : MENU 1,3,1 : MENU 1,2,0 :
 MENU 1,4,0
  132
  133  RETURN
  134
  135  Markensetzen:
  136
  137    CIRCLE (170,150),20,2 : CIRCLE (320,150),20,2 :CIRC
LE (470,150),20,2
  138    PAINT (170,150),2     : PAINT (320,150),2     :PAIN
T (470,150),2
  139
  140    CIRCLE (95,110),20,2 : CIRCLE (245,110),20,2 :CIRCL
E (395,110),20,2
  141    CIRCLE (545,110),20,2
  142    PAINT (95,110),2      : PAINT (245,110),2     :PAINT
 (395,110),2
  143    PAINT (545,110),2
  144
  145    CIRCLE (170,70),20,2 : CIRCLE (320,70),20,2 :CIRCLE
 (470,70),20,2
  146    PAINT (170,70),2      : PAINT (320,70),2      :PAINT
(470,70),2
  147
  148    CIRCLE (245,30),20,2 :CIRCLE (395,30),20,2
  149    PAINT (245,30),2      :PAINT (395,30),2
  150
  151    MENU 1,2,0 : MENU 1,3,0 : MENU 1,4,1 : MENU 1,1,0 :
 MENU 1,5,0
  152
  153  RETURN
  154
  155  Zweispieler:
  156
  157    WINDOW 3,"Spieler: ",(0,10) - (200,30),0,2
  158    SOUND WAIT
  159    SOUND 261.63,15,127,0 : SOUND 329.63,15,96,1 : SOUN
D 392!,15,96,2
  160    SOUND RESUME
  161    INPUT "Spieler 1: "; spieler1$
  162    SOUND WAIT
  163    SOUND 523.25,15,127,0 : SOUND 659.26,15,96,1 : SOUN
D 783.99,15,96,2
  164    SOUND RESUME
  165    INPUT "Spieler 2: "; spieler2$
  166    MENU 1,2,1
  167    WINDOW CLOSE 3
  168
  169  RETURN
  170
  171  Spielereintrag:
  172
  173    LOCATE 3,10 : PRINT "Spieler 1:" : LOCATE 5,10 : PR
INT USING "\          \"; spieler1$
  174    LOCATE 3,60 : PRINT "Spieler 2:" : LOCATE 5,60 : PR
INT USING "\          \"; spieler2$
  175    IF spieler1$="" THEN player1$="Spieler 1" ELSE play
er1$=spieler1$
  176    IF spieler2$="" THEN player2$="Spieler 2" ELSE play
er2$=spieler2$
  177
  178  RETURN
  179
  180  Menuauswahl:
  181
  182    men = MENU(0) : menupunkt = MENU(1)
  183    Abbruch% = 1
  184
  185  RETURN
  186
  187  InfoAnzeige:
  188
  189    men=MENU(0)
  190    menupunkt=MENU(1)
  191    IF men = 1 AND (menupunkt = 1 OR menupunkt = 6) THE
N i% = 0
  192    GOSUB Wahl1
  193
  194  RETURN
  195
  196  Spielregeln:
  197
  198    CLS
  199    PRINT
  200    PRINT "    D I E   S P I E L R E G E L N   V O N
 N I M B I :"
  201    PRINT : PRINT
  202    PRINT "   Nimbi ist ein Strategiespiel, das von 2 P
ersonen gespielt wird. "
  203    PRINT "   Die Spieler entfernen abwechselnd einen o
der mehrere der schwarzen "
  204    PRINT "   Steine vom Spielfeld, bis kein Stein mehr
 auf dem Spielfeld vorhanden "
  205    PRINT "   ist. Mehrere Steine können nur dann entfe
rnt werden, wenn sie auf "
  206    PRINT "   einer geraden Linie liegen und eine ununt
erbrochene Kette bilden, so "
  207    PRINT "   daß kein leeres Feld dazwischen liegt."
  208    PRINT
  209    PRINT "   VERLIERER ist, wer den letzten Stein entf
ernen muß. "
```

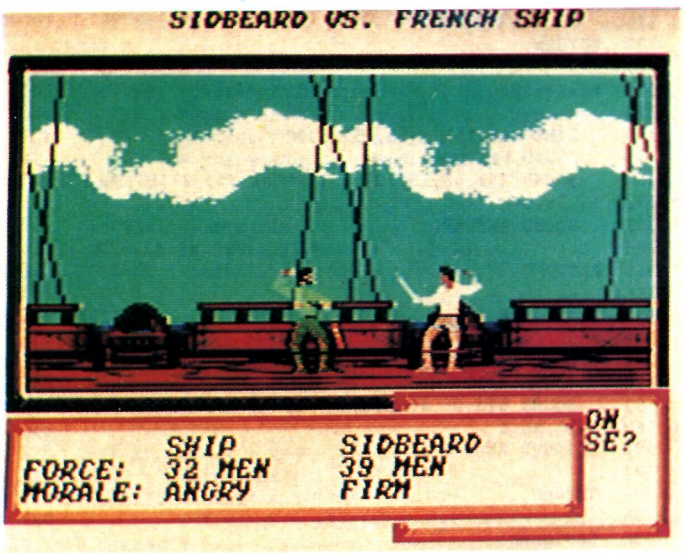
```
  210    PRINT : PRINT
  211    PRINT "   Im Programm sind die Spielfelder orangefa
rben, die Spielsteine schwarz "
  212    PRINT "   markiert. "
  213    PRINT "   Das Entfernen der Steine geschieht in der
 Weise, daß man mit der linken "
  214    PRINT "   Maustaste die Steine, die man entfernen m
öchte, markiert (Sie sind "
  215    PRINT "   dann blau dargestellt) und anschließend i
m Menu 'Marken' den Menupunkt "
  216    PRINT "   'Entfernen' wählt. Durch die Wahl von 'Un
do' kann eine Markierung rÜck- "
  217    PRINT "   gängig gemacht werden. "
  218
  219    vonneuem% = 1 : MENU 1,1,1 : MENU 1,2,0 : MENU 1,3,
0 : MENU 1,4,0 : MENU 1,5,0
  220
  221  RETURN
  222
  223  Spielende:
  224
  225    SYSTEM
  226
  227  RETURN
  228
  229  Checkmenu:
  230
  231      men = MENU(0)
  232      menupunkt = MENU(1)
  233      ON men GOSUB Wahl1, Wahl2
  234
  235  RETURN
  236
  237  Wahl1:
  238
  239    ON menupunkt GOSUB Spielfeld,Starten,Zweispieler,Ne
ubeginn,Spielregeln,Spielende
  240
  241  RETURN
  242
  243  Wahl2:
  244
  245    ON menupunkt GOSUB Checkmark,Undo
  246
  247  RETURN
  248
  249  Starten:
  250
  251    GOTO Spielstart
  252
  253  RETURN
  254
  255  Menuleiste:
  256
  257    MENU 1,0,1,"Auswahl"
  258    MENU 1,1,1,"Spielfeld"
  259    MENU 1,2,0,"Spielstart"
  260    MENU 1,3,0,"Spieler: "
  261    MENU 1,4,0,"Neubeginn"
  262    MENU 1,5,1,"Spielregeln"
  263    MENU 1,6,1,"Quit"
  264
  265
  266    MENU 2,0,1,"Marken"
  267    MENU 2,1,0,"Entfernen"
  268    MENU 2,2,0,"Undo"
  269
  270    MENU 3,0,0," "
  271
  272    MENU 4,0,0," "
  273
  274  RETURN
  275
  276  Undo:
  277
  278    FOR k% = 0 TO 4
  279      FOR n% = 0 TO k%
  280      FOR m% = 0 TO (k%-n%)
  281        IF mark%(n%,m%)=1 THEN mark%(n%,m%)=0 :CIRCLE (
xpos%(n%,m%),ypos%(n%,m%)),20,2 : PAINT(xpos%(n%,m%),ypos%(n
%,m%)),2
  282    NEXT m%,n%,k%
  283    MENU 2,2,0 : MENU 2,1,0 :stzahl% = 0
  284
  285  RETURN
  286
  287  Steinmark:
  288
  289    gut=MOUSE(0)
  290
  291    IF MOUSE(1)<20 AND MOUSE(2)<10 THEN RETURN
  292
  293    FOR k=0 TO 4
  294    FOR i=0 TO k
  295    FOR j=0 TO (k-i)
  296      IF ABS(MOUSE(1)-xpos%(i,j))<18 AND ABS(MOUSE(2)-y
pos%(i,j))<6 THEN
  297        IF mark%(i,j)=0 THEN GOSUB Hilfsroutine
  298      END IF
  299    NEXT j,i,k
  300    IF stzahl% = 1 THEN MENU 2,2,1 : MENU 2,1,1
  301
  302  RETURN
  303
  304  Hilfsroutine:
  305
  306    mark%(i,j) = 1
  307    IF stzahl% = 0 THEN xmin% = i : ymin% = j : xmax% =
 i : ymax% = j
  308    IF stzahl% > 0 THEN
  309      IF i < xmin% THEN xmin% = i : ymin% = j
  310      IF i = xmin% AND j < ymin% THEN ymin% = j
  311      IF i > xmax% THEN xmax% = i : ymax% = j
  312      IF i = xmax% AND j > ymax% THEN ymax% = j
  313    END IF
  314
  315    stzahl% = stzahl%+1
  316    CIRCLE (xpos%(i,j),ypos%(i,j)),20,6
  317    PAINT (xpos%(i,j),ypos%(i,j)),6
  318
  319  RETURN
  320
  321  Korrektemark:
  322
  323    correct% = 0
  324
  325    IF stzahl% = 1 THEN
  326            correct% = 1
  327    ELSEIF stzahl% < 5 THEN
  328            IF xmin% = xmax% AND (ymin%+stzahl%-1) = ym
ax% THEN correct% = 1
  329            IF ymin% = ymax% AND (xmin%+stzahl%-1) = xm
ax% THEN correct% = 1
  330            IF (xmin%+stzahl%-1) = xmax% AND (ymin%-stz
ahl%+1) = ymax% THEN GOSUB Summentest
  331    ELSE
  332            correct% = 0
  333    END IF
  334
  335  RETURN

  336
  337  Summentest:
  338
  339    diagonale% = xmin%+ymin% : correct% = 1
  340
  341      FOR i% = 0 TO diagonale%
  342        FOR j% = 0 TO diagonale%-i%
  343          IF mark%(i%,j%) = 1 AND i%+j% <> diagonale% T
HEN correct% = correct%*0
  344    NEXT j%,i%
  345
  346  RETURN
  347
  348  Checkmark:
  349
  350    GOSUB Korrektemark
  351    IF correct% = 1 THEN GOSUB Markenentfernen ELSE GOS
UB Undo
  352
  353  RETURN
  354
  355  Markenentfernen:
  356
  357    h%=0
  358    FOR k=0 TO 4
```

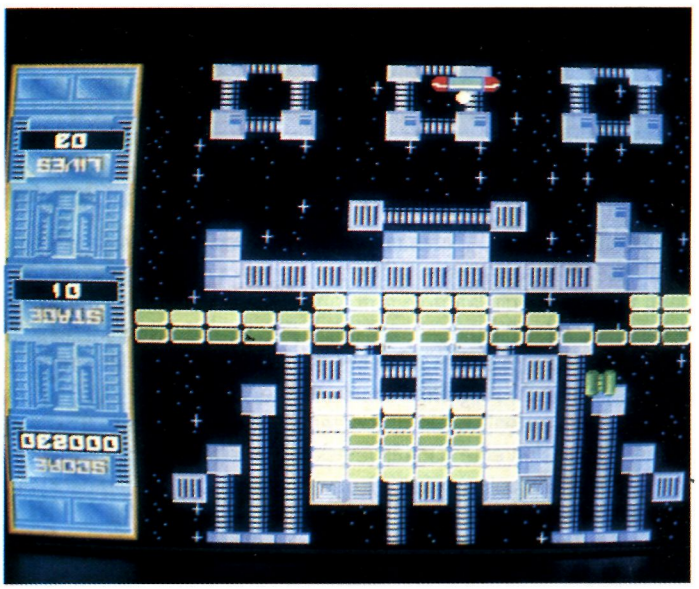
```
  359    FOR i=0 TO k
  360    FOR j=0 TO (k-i)
  361      IF mark%(i,j)=1 THEN h%=h%+1 : CIRCLE (xpos%(i,j)
,ypos%(i,j)),20,3 : PAINT (xpos%(i,j),ypos%(i,j)),3 : mark%(
i,j)=2
  362    NEXT j,i,k
  363    anzmarken% = anzmarken% - h%
  364    IF reihenf% = 1 THEN reihenf% = 2 :LINE (35,15) -(5
5,25),0,BF : GOSUB ZweiterSP ELSE reihenf% = 1: LINE (570,15
) - (590,25),0,BF : GOSUB ErsterSp
  365    test% = 2 : MENU 2,1,0 : MENU 2,2,0
  366
  367
  368  RETURN
  369
  370  Neubeginn:
  371
  372    MENU 1,1,0 : MENU 1,3,1 : MENU 1,4,0 : MENU 1,5,1:
MENU 1,2,1 : MENU 2,1,0 : MENU 2,2,0
  373    IF vonneuem% = 1 THEN CLS :MENU 1,1,1 : MENU 1,2,0
: MENU 1,3,0 : GOSUB Spielfeld
  374    GOTO Hauptschleife
  375
  376  RETURN
  377
  378  ErsterSp:
  379
  380    LINE (35,15) -(55,25),5,BF
  381    SOUND WAIT
  382    SOUND 261.63,15,127,0 : SOUND 349.23,15,96,1 : SOUN
D 392!,15,96,2
  383    SOUND RESUME
  384
  385  RETURN
  386
  387  ZweiterSP:
  388
  389    LINE (570,15) - (590,25),5,BF
  390    SOUND WAIT
  391    SOUND 523.25,15,127,0 : SOUND 659.26,15,96,1 : SOUN
D 587.33,15,96,2
  392    SOUND RESUME
  393
  394  RETURN
  395
  396  Gewinner:
  397
  398    WINDOW 4,"Gewinner",(10,150) - (130,180),0,2
  399    LOCATE 3,2
  400    IF reihenf% = 1 THEN PRINT player1$ ELSE PRINT play
er2$
  401    MENU 1,4,1 : MENU 2,2,0 : MENU 2,1,0 : MENU 1,1,0 :
MENU 1,2,0 : MENU 1,3,0 : MENU 1,5,0
  402    i%=1
  403    WHILE i% <> 0
  404    ON MENU GOSUB Weiter
  405    MENU ON
  406    WEND
  407
  408  RETURN
  409
  410  Weiter:
  411
  412    WINDOW CLOSE 4
  413    vonneuem% = 0
  414    LINE (35,15) - (55,25),0,BF : LINE (570,15) - (590,
25),0,BF
  415    GOSUB Checkmenu
  416
  417  RETURN
  418
  419  Wertefestlegung:
  420
  421    FOR n%=0 TO 4
  422      FOR m%=0 TO 4
  423        mark%(n%,m%)=0
  424    NEXT m%,n%
  425
  426    FOR k% = 0 TO 4
  427      FOR n% = 0 TO k%
  428      FOR m% = 0 TO (k% - n%)
  429        IF n% = 0 AND m% = 0 THEN xpos%(n%,m%) = 0 : yp
os%(n%,m%) = 0
  430        IF n% = 4 OR m% = 4 THEN xpos%(n%,m%) = 0 : ypo
s%(n%,m%) = 0
  431        IF (n%>0 OR m%>0) AND n%<4 AND m%<4 THEN xpos%(
n%,m%) = 20+n%*150+m%*75 : ypos%(n%,m%) = 150-m%*40
  432
  433    NEXT m%,n%,k%
  434
  435  RETURN
  436
  437  Nullsetzen:
  438
  439    FOR k% = 0 TO 4
  440      FOR n% = 0 TO k%
  441      FOR m% = 0 TO (k%-n%)
  442        IF mark%(n%,m%) > 0 THEN mark%(n%,m%)=0
  443    NEXT m%,n%,k%
  444
  445  RETURN
  446
```


## Ihr Programm in Roeske's Computer Mag.

Sie haben bereits ein Programm geschrieben, von dem Sie annehmen, daß es auch andere Leser interessiert? Sie würden es gerne gegen Honorar veröffentlichen?
Dann sollten Sie nicht lange zögern. Rufen Sie uns an. Nennen Sie uns Ihre Telefonnummer und wir rufen Sie umgehend zurück (Um Ihren Geldbeutel zu entlasten!). Wir werden Ihnen alles mitteilen, was Sie wissen wollen, wir reden übers finanzielle und Sie erzählen uns, was Ihr Programm kann, und wie Sie sich eine Veröffentlichung vorstellen.
Als nächstes werden Sie uns Ihr Programm vorstellen wollen. Zu diesem Zweck senden Sie uns am besten eine gut ausgetestete Diskette zu, damit wir uns ein Urteil bilden können und dann entscheiden, ob das Programm in einer der nächsten Ausgaben vorgestellt werden soll.
Jetzt ist der Zeitpunkt gekommen, zu dem wir Ihnen unsere Entscheidung mitteilen und uns auf gute Zusammenarbeit freuen.
So können Sie als Leser Ihr Magazin mitgestalten.
Wir veröffentlichen Programme in Assembler, Basic und C. Von Programmen, die wegen ihrer Länge den Rahmen des Magazins sprengen würden, profitieren unsere Leser (und Sie) durch unseren Disketten-Service.

Wir freuen uns auf Ihren Anruf unter der Telefon-Nummer.: 05651-8778

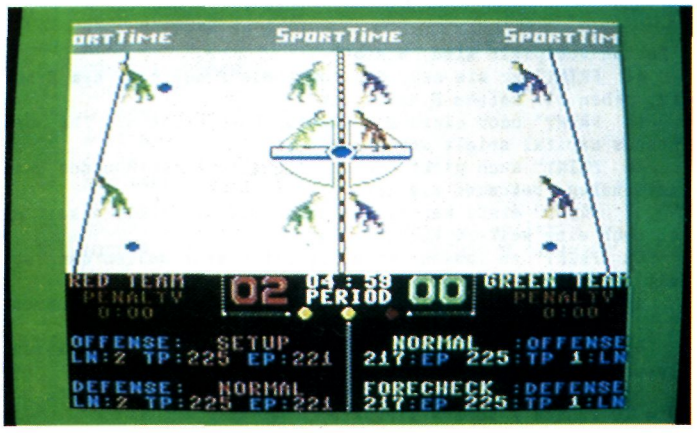


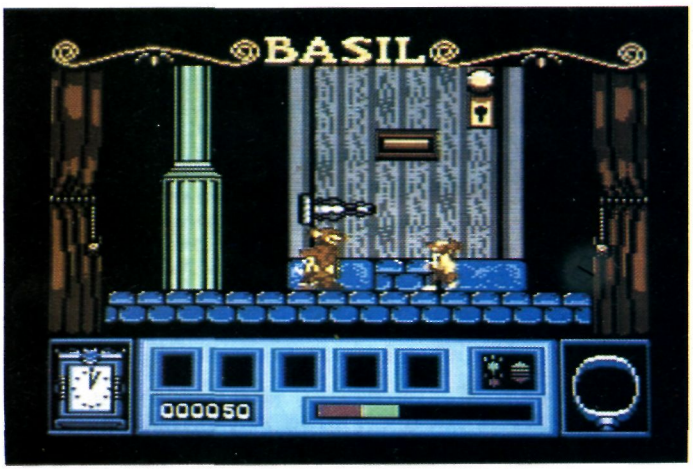


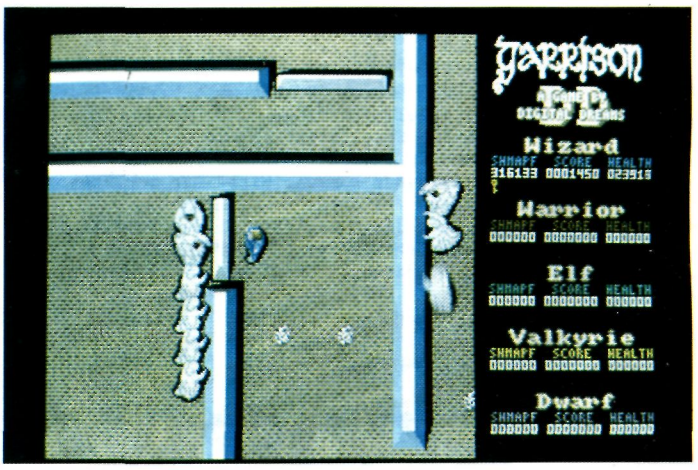
**Hotline**

**☎ 05651-8778**

# *Roeske's Computer Mag.* *Gewinnspiel*





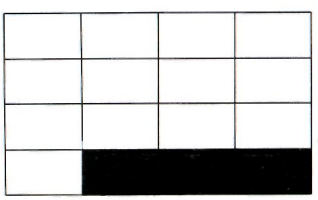




**Rainbow Arts** und **Roeske's Computer Mag.** veranstalten in dieser Ausgabe ein kleines Gewinnspiel, bei dem Sie zeigen müssen, wie gut Sie sich in der Homecomputer-Szene auskennen. Füllen Sie bitte nebenstehenden Cupon aus und senden Sie ihn mit Ihrer Adresse an:

Roeske's Computer Mag.
Westring 59c
3440 Eschwege

Einsendeschluß ist der **29. Februar 1988** - Der Rechtsweg ist ausgeschlossen.

Unsere Frage lautet: Welches der abgebildeten Bildschirmfotos zeigt ein Spiel von **Rainbow Arts**. Wenn sie eines oder mehrere entdeckt haben, dann machen Sie ein Kreuz im nebenstehenden Raster an der entsprechenden Position.



Aus den richtigen Einsendungen losen wir 30 Gewinner aus, die Amiga- oder C64-Spiele von Rainbow Arts überreicht bekommen.
Bitte beantworten Sie uns noch folgende Fragen:
Welches System besitzen Sie? ☐ Amiga ☐ C64 ☐ Sonstiges
Was bevorzugen Sie? ☐ Listings ☐ Reviews ☐ Tips & Tricks
Wie gefällt Ihnen RCM? ☐ Sehr gut ☐ Es geht ☐ Gar nicht
Wie lange computern Sie schon? ☐ 0-1 Jahr ☐ 1-2 Jahre ☐ länger

## 17 + 4 Das bekannte Kartenspiel - gegen den Amiga!

```
    1  CLEAR
    2  RANDOMIZE TIMER
    3  DIM Pik%(40):DIM Kreuz%(40):DIM Herz%(40):DIM Karo%(4
0):DIM card(32)
    4  SCREEN 1,640,200,3,2
    5  WINDOW 1," 17 + 4  von Heinz Prangenberg  (Roeske's C
omputer Mag. 1/88",,16,1
    6  PALETTE 0,1,1,1
    7  PALETTE 1,0,0,0
    8  PALETTE 2,.93,0,0
    9  PALETTE 3,.33,.87,0
   10  COLOR 1
   11  AREA (0,13):zz=18:SOUND 700,20,200:GOSUB Areas
   12  AREA (578,30):zz=15:GOSUB Areas:RESTORE:GOSUB Lauf
   13  AREA (57,23):zz=18:SOUND 850,20,200:GOSUB Areas
   14  AREA (506,40):zz=15:GOSUB Areas:RESTORE:GOSUB Lauf
   15  AREA (114,33):zz=18:SOUND 1000,20,200:GOSUB Areas
   16  AREA (434,50):zz=15:GOSUB Areas:RESTORE:GOSUB Lauf
   17  AREA (171,43):zz=18:SOUND 1150,20,200:GOSUB Areas
   18  AREA (362,60):zz=15:GOSUB Areas:RESTORE:GOSUB Lauf
   19  COLOR 2
   20  AREA (229,53):zz=18:GOSUB Areas
   21  AREA (289,70):zz=15:GOSUB Areas:SOUND 1300,20,255:GOT
O Beschreibung
   22  Areas:
   23  FOR i=1 TO zz:READ a,b
   24  AREA STEP (a,b):NEXT i
   25  AREAFILL :RETURN
   26  DATA 14,-11,48,0,-9,32,6,0,0,8,-8,0,-6,15
   27  DATA -14,0,5,-15,-4,0,0,-8,6,0,7,-22,-18,0
   28  DATA 0,45,-16,0,0,-38,-11,7
   29  DATA 10,0,0,4,6,0,8,-32,16,0,-8,32,7,0,0,-7
   30  DATA 13,0,0,30,-13,0,0,-15,-29,0,0,4,-10,0
   31  Beschreibung:
   32  COLOR 1:LOCATE 15,27:PRINT"Spielerklaerung (J/N) ?"
   33  T3:
   34  a$=INKEY$:IF a$="" THEN T3
   35  IF a$="n" THEN Eingabe
   36  IF a$<>"j" THEN BEEP:GOTO T3
   37  COLOR 1,3:CLS
   38  LOCATE 1,30:PRINT"Spielbeschreibung":PRINT
   39  PRINT" Da dieses Spiel ja allgemein bekannt ist, nur
ein paar Worte zum Spielverlauf."
   40  PRINT" Ziel des Spieles ist es, so nahe wie moeglich
an 21 Punkte zukommen. Sollten"
   41  PRINT" Sie darueber kommen, haben Sie verloren. Zu Be
ginn des Spieles fragt der Amiga"
   42  PRINT" Sie nach Ihrem Kapital, ohne spielt er nicht.
Nun mischt er die Karten, es"
   43  PRINT" sind 32 davon im Spiel und immer wenn diese 32
 Karten verspielt sind, mischt"
   44  PRINT" er sie neu. Nun kommt die Frage nach dem Einsa
tz, geben Sie keinen Einsatz ein"
   45  PRINT" oder einen Einsatz der hoeher ist als Ihr mome
ntanes Kapital spielt der Amiga"
   46  PRINT" auch nicht. Sollte Sie alle diese Huerden geno
mmen haben, bekommen Sie Ihre"
   47  PRINT" erste Karte. Nun haben Sie die Moeglichkeit si
ch noch eine weitere Karte geben"
   48  PRINT" zu lassen, wenn Sie keine mehr wollen dann spi
elt der Amiga. Gewonnen hat der,"
   49  PRINT" der am naechsten an die Zahl 21 herangekommen
ist. Bei Punktgleichheit gewinnt"
   50  PRINT" der Computer. Die Karten sind folgendermassen
bepunktet :"
   51  PRINT" 7 = 7 Punkte / 8 = 8 Punkte / 9 = 9 Punkte / 1
0 = 10 Punkte / Bube = 2 Punkte "
   52  PRINT" Dame = 3 Punkte / Koenig = 4 Punkte / As = 11
Punkte / As+As = 21 Punkte."
   53  PRINT" Ist die Erste Karte ein As und die zweite Kart
e auch ein As haben Sie 21 Pkt."
   54  PRINT" Nun muesste alles gesagt sein.  Ich wuensche
Ihnen viel Spass."
   55  LOCATE 22,20:PRINT"Zum Programmstart bitte die SPACET
ASTE druecken"
   56  T4:
   57  a$=INKEY$:IF a$="" THEN T4
   58  IF a$<>" " THEN BEEP:GOTO T4
   59  Eingabe:
   60  GOSUB Datalesen
   61  COLOR 1,0:CLS:zz=0:a=0:b=0
   62  LOCATE 10,10:PRINT"Sie wollen also gegen mich spielen
,"
   63  LOCATE 12,10:INPUT"Wieviel Spielkapital haben Sie den
n ";kapital
   64  IF kapital<1 THEN Eingabe
   65  COLOR 2:LOCATE 15,20:PRINT"Na dann wollen wir mal !!"
   66  FOR i=1 TO 4000:NEXT i
   67  LINE (0,0)-(472,144),3,bf:COLOR 1,0
   68  LOCATE 20,3:PRINT "Ihre Punkte:    Meine Punkte:    K
apital:    Ihr Einsatz:"
   69  Start:
   70  karte=32
   71  IF kapital<1 THEN Ende
   72  COLOR 0,2
   73  LOCATE 10,62:PRINT" Moment bitte "
   74  LOCATE 12,62:PRINT"  ich mische  "
   75  LOCATE 14,62:PRINT"  die Karten  "
   76  FOR i=1 TO 300
   77  Zuf:
```

CAS-COMPUTER Geschäftsführer: A. Dedecke u. B. Stevermüer

2 MB für Amiga 500/1000/2000 DM 948,-

Vertrieb Schweiz: Mega - Shop · Müllerstr. 64 · 8004 Zürich · Tel. 01/24 17 101
Vertrieb BRD: Take Over Microcomputer · Kurze Geismarstr. 21 · 3400 Göttingen · Tel. 05 51/48 50 89
Kelkel Computershop · Seidelbastring 9 · 6490 Schlüchtern · Tel. 0 66 61/25 71
Mailbox: 0 61 88/64 87
Bankverbindungen: Bezirks-Sparkasse Seligenstadt BLZ 506 521 24 · Konto-Nr. 14 10 10 91 · Postgiroamt Frankfurt Konto-Nr. 469998-609 A. Dedecke

Sprendlinger Landstraße 71 · 6050 Offenbach · Ruf (0 69) 84 20 13

```
   78  a=INT(RND*(33)):IF a<1 THEN Zuf
   79  Zufa:
   80  b=INT(RND*(33)):IF b<1 OR b=a THEN Zufa
   81  SWAP card(a),card(b)
   82  NEXT i
   83  GOSUB loeschen
   84  COLOR 1,0
   85  LOCATE 22,35:PRINT kapital;"DM"
   86  Routine:
   87  IF kapital<1 THEN Ende
   88  ihrpunkt=0:punkt=0
   89  Schleife:
   90  COLOR 1,0:LOCATE 6,62:PRINT"Wieviel wollen"
   91  LOCATE 7,62:PRINT"Sie setzen":LOCATE 9,62:INPUT einsa
tz
   92  IF einsatz<1 OR einsatz>kapital THEN Schleife
   93  GOSUB loeschen
   94  kapital=kapital-einsatz
   95  COLOR 1,0
   96  LOCATE 22,35:PRINT kapital;"DM   "
   97  LOCATE 22,50:PRINT einsatz;"DM  "
   98  LINE (16,17)-(62,64),0,bf
   99  karte=karte-1:GOSUB Bildzeigen
  100  x=6:y=4:xx=24:GOSUB Anzeigen
  101  waag=16
  102  COLOR 1,0:LOCATE 22,5:PRINT punkt:LOCATE 22,22:PRINT
ihrpunkt
  103  GOTO Abfrage
  104  Bildzeigen:
  105  kb=INT(card(karte)/100)
  106  IF fb<3 THEN kf=2 ELSE kf=1
  107  pz=card(karte)-(kb*100)
  108  RETURN
  109  Anzeigen:
  110  COLOR kf,0
  111  IF kb=1 THEN PUT (y*8-8,xx),Karo%
  112  IF kb=2 THEN PUT (y*8-8,xx),Herz%
  113  IF kb=3 THEN PUT (y*8-8,xx),Pik%
  114  IF kb=4 THEN PUT (y*8-8,xx),Kreuz%
  115  IF pz<7 OR pz>10 THEN AnzeigenB
  116  LOCATE x,y-1:PRINT pz:punkt=punkt+pz
  117  RETURN
  118  AnzeigenB:
  119  IF pz=2 THEN b$="B"
  120  IF pz=3 THEN b$="D"
  121  IF pz=4 THEN b$="K"
  122  IF pz=11 AND y=4 THEN b$="A":ass=1
  123  IF pz=11 AND y<>4 THEN b$="A"
  124  LOCATE x,y:PRINT b$:punkt=punkt+pz
  125  RETURN
  126  Nochkarte:
  127  waag=waag+48:y=y+6
  128  LINE (waag,17)-(waag+46,64),0,bf
  129  karte=karte-1
  130  GOSUB Bildzeigen
  131  GOSUB Anzeigen
  132  COLOR 1,0
  133  IF pz=11 AND ass=1 THEN punkt=21:ass=0
  134  ass=0
  135  LOCATE 22,5:PRINT punkt
  136  IF punkt>21 THEN Zuviel
  137  GOTO Abfrage
  138  KeineKarte:
  139  ihrpunkt=punkt:punkt=0
  140  COLOR 1,3
  141  waag=-32:y=-2:x=14:xx=88:v=0:ass=0
  142  NKarte:
  143  COLOR 1,3:v=v+1
  144  waag=waag+48:y=y+6:
  145  LINE(waag,81)-(waag+46,128),0,bf
  146  karte=karte-1
  147  GOSUB Bildzeigen
  148  GOSUB Anzeigen
  149  COLOR 1,0
  150  IF pz=11 AND ass=1 AND v=2  AND y<>4 THEN punkt=21:as
s=0
  151  LOCATE 22,22:PRINT punkt
  152  IF ihrpunkt>punkt THEN FOR i=1 TO 3000:NEXT i:GOTO NK
arte
  153  GOTO Auswertung
  154  Zuviel:
  155  GOSUB loeschen
  156  COLOR 1,2
  157  LOCATE 4,62:PRINT" Das war eine "
  158  LOCATE 6,62:PRINT" Karte zuviel "
  159  GOTO Loeschen2
  160  Auswertung:
  161  COLOR 1,2
  162  IF ihrpunkt>punkt OR punkt>21 THEN
  163  LOCATE 3,61:PRINT"                    "
  164  LOCATE 4,61:PRINT" Sie haben gewonnen "
  165  LOCATE 5,61:PRINT"                    "
  166  kapital=kapital+einsatz*2:COLOR 1,0:LOCATE 22,35:PRIN
T"              "
  167  LOCATE 22,35:PRINT kapital;"DM"
  168  END IF
  169  IF punkt<=21 AND ihrpunkt<=punkt THEN
  170  LOCATE 4,62:PRINT" Sie haben "
  171  LOCATE 6,62:PRINT" verloren "
  172  END IF
  173  GOTO Loeschen2
  174  Datalesen:
  175  FOR i=0 TO 10:READ Pik%(i):NEXT i
  176  DATA 16,8,2
  177  DATA 384,960,4080,16380,32766,14748,384,960
  178  FOR i=0 TO 10:READ Kreuz%(i):NEXT i
  179  DATA 16,8,2
  180  DATA 2016,2016,29070,32766,29070,384,384,960
  181  FOR i=0 TO 18:READ Herz%(i):NEXT i
  182  DATA 16,8,2
  183  DATA 0,0,0,0,0,0,0,0
  184  DATA 14364,-385,-1,-1,16380,4080,960,384
  185  FOR i=0 TO 18:READ Karo%(i):NEXT i
  186  DATA 16,8,2
  187  DATA 0,0,0,0,0,0,0,0
  188  DATA 384,2016,8184,32766,8184,2016,384,0
  189  FOR i=1 TO 32:READ card(i):NEXT i
  190  DATA 107,108,109,110,111,102,103,104
  191  DATA 207,208,209,210,211,202,203,204
  192  DATA 307,308,309,310,311,302,303,304
  193  DATA 407,408,409,410,411,402,403,404
  194  RETURN
  195  Abfrage:
  196  COLOR 3,1
  197  LOCATE 4,62:PRINT" Noch eine Karte "
  198  LOCATE 6,67:PRINT" (j/n) "
  199  T1:
  200  a$=INKEY$:IF a$="" THEN T1
  201  IF a$="n" THEN GOSUB loeschen:GOTO KeineKarte
  202  IF a$<>"j" THEN BEEP:GOTO T1
  203  GOTO Nochkarte
  204  loeschen:
  205  COLOR 0,0
  206  FOR i=1 TO 18
  207  LOCATE i,61:PRINT"                    "
  208  NEXT i:RETURN
  209  Loeschen2:
  210  COLOR 1,0
  211  einsatz=0:LOCATE 22,47:PRINT"
  "
  212  LOCATE 22,50:PRINT einsatz;"DM"
  213  COLOR 2,1
  214  LOCATE 10,62:PRINT" Noch ein Spiel "
  215  LOCATE 12,65:PRINT" (J/N) "
  216  T2:
  217  a$=INKEY$:IF a$="" THEN T2
  218  IF a$="j" AND karte<15 THEN LINE(0,0)-(472,144),3,bf:
GOSUB K:GOSUB loeschen:ass=0:GOTO Start
  219  IF a$="j" AND karte>=15 THEN LINE(0,0)-(472,144),3,bf
:GOSUB K:GOSUB loeschen:ass=0:GOTO Routine
  220  IF a$<>"n" THEN BEEP:GOTO T2
  221  LINE (0,0)-(472,144),3,bf:GOSUB loeschen:COLOR 2,1
  222  LOCATE 10,2:PRINT" Schade dass Sie keine Lust mehr ha
ben. "
  223  LOCATE 12,2:PRINT" Ich hoffe es hat Ihnen Spass gemac
ht. "
  224  LOCATE 14,2:PRINT" Bis zum naechsten mal !! ":PRINT:P
RINT:END
  225  K:
  226  IF kapital<=0 THEN LOCATE 10,2:PRINT" Das Spiel ist b
eendet. Sie haben kein Geld mehr !! "
  227  IF kapital<=0 THEN LOCATE 12,2:PRINT" Es hat mich seh
r gefreut, mit Ihnen zu spielen. ":PRINT:PRINT:END
  228  RETURN
  229  Lauf:
  230  FOR i=1 TO 1500:NEXT i:CLS:RETURN
```

# Das Börsenspiel

**Spekulieren Sie gern? Dann ist das Börsenspiel genau richtig für Sie. Der Amiga stellt Ihnen Bargeld zur Verfügung, verwaltet Ihre Konten und führt die Transaktionen durch, die Sie wünschen.**
**Nebenbei wickelt er auch noch sämtliche Börsengeschäfte ab, ist also schwer beschäftigt, Sie zu unterhalten.**

**Achtung! Alle Amiga-Listings bitte ohne Zeilennummern abtippen.**

```
    1  DEFINT a-Z
    2  CLEAR 30000
    3  Farbe=1
    4  a=2:b=10:G=5000:K=5000:Bq=20:R=.5:Ewertu=5:Ewertk=105
:yy=1:KF=5000
    5  DIM WW(10)
    6  DIM AK$(b),We(b),VE(b),an(b),Kk(b),Gg(b),Te$(b),Pp(10
,5)
    7  FOR t=1 TO 10:We(t)=300:Te$(t)="?":VE(t)=0:NEXT t
    8  Bildaufbau:
    9  SCREEN 1,640,200,3,2
   10  WINDOW 1,"BOERSE  von Heinz Prangenberg  (c) 1988 RCM
 Eschwege",,16,1
   11  CLS:LOCATE 10,27:PRINT"B O E R S E N S P I E L"
   12  LOCATE 12,27:PRINT"Spielbeschreibung  (J/N)"
   13  Beschreibung:
   14  a$=INKEY$:IF a$="" THEN Beschreibung
   15  IF a$="n" THEN Musikbeginn
   16  IF a$<>"j" THEN BEEP:GOTO Beschreibung
   17  CLS:COLOR 3,0:LOCATE 1,30:PRINT"Spielbeschreibung"
   18  PRINT" Das Spiel Boerse ermoeglicht es Ihnen wie ein
Boersenmakler an der Boerse"
   19  PRINT" zu spekulieren. Sie haben zu Beginn des Spiele
s 5000 DM Spielkapital zur"
   20  PRINT" Verfuegung. Im Fenster Bargeld ist immer der B
etrag den Sie bar zur Ver -"
   21  PRINT" fuegung haben angezeigt. Im Fenster GesamtKapi
tal ist der Betrag angezeigt"
   22  PRINT" den Sie besitzen wuerden wenn Sie Ihren gesamt
en Aktienbesitz in diesem"
   23  PRINT" Moment verkaufen wuerden plus Bargeld. Wenn Si
e in der Menueleiste die"
   24  PRINT" Punkte kaufen oder verkaufen anwaehlen, werden
 die Firmen angezeigt von"
   25  PRINT" denen Sie Aktien erwerben koennen. Haben Sie e
ine Firma angewaehlt, bekommen"
   26  PRINT" Sie im Fenster Taetigkeiten die Frage gestellt
 wieviele Aktien Sie von"
   27  PRINT" dieser Firma kaufen oder verkaufen moechten. I
n dem unteren Fenster Besitz"
   28  PRINT" werden Ihnen die Firmen angezeigt, Ihr momenta
ner Aktienbesitz und der Preis"
   29  PRINT" jeder Aktie. Wollen Sie neue Aktienkurse, mues
sen Sie in der Menueleiste den"
   30  PRINT" Punkt Weiter anwaehlen, das geht aber nur wenn
 Sie mindestens einen Kauf"
   31  PRINT" oder Verkauf durchgefuehrt haben. Weiterhin ha
ben Sie in der Menueleiste"
   32  PRINT" durch den Punkt Tendenz die moeglichkeit sich
anzeigen zu lassen welche"
   33  PRINT" Aktien sich seid dem letzten Kurs veraendert h
aben und um vieviel DM sich"
   34  PRINT" der Preis verschoben hat. Unter Punkt Ende koe
nnen Sie das Programm beenden"
   35  PRINT" und Sie bekommen eine Ausgabe darueber ob Sie
Gewinn oder Verlust gemacht"
   36  PRINT" haben.          Und nun wuensche ich Ihnen vie
l Glueck an der Boerse"
   37  COLOR 1,0
   38  PRINT:PRINT"                    Zum Programmstart bitte
 SPACETASTE druecken"
   39  Spacetaste:
   40  a$=INKEY$:IF a$="" THEN Spacetaste
   41  IF a$<>" " THEN BEEP:GOTO Spacetaste
   42  Musikbeginn:
   43  READ Freq,Dauer
   44  IF Freq=-1 THEN Spielbeginn
   45  SOUND WAIT
   46  SOUND Freq,Dauer,21,0
   47  SOUND Freq/2,Dauer,127,1
   48  SOUND Freq*2,Dauer,21,2
   49  GOTO Musikbeginn
   50  DATA 523.25,15,784,15,698.48,6
   51  DATA 659.28,6,587.28,6,1046.52,15
   52  DATA 784,17,-1,-1
   53  Spielbeginn:
   54  CLS
   55  WINDOW 2,"Bargeld",(5,15)-(100,70),16,1
   56  WINDOW 3,"Besitz",(5,150)-(630,180),16,1
   57  WINDOW 4,"Gesamkapital",(115,15)-(210,70),16,1
   58  WINDOW 5,"Taetigkeiten",(225,15)-(330,70),16,1
   59  RANDOMIZE TIMER
   60  PALETTE 0,0,0,0:PALETTE 4,.6,.1,.9
   61  PALETTE 1,.3,.3,1:PALETTE 5,.6,.6,.6
   62  PALETTE 2,.7,.6,.4:PALETTE 6,.1,.9,.5
   63  PALETTE 3,.4,.7,.8:PALETTE 7,.8,.4,.3
   64  GOSUB Aufbau
   65  MENU 1,0,1,"Weiter"
   66  MENU 1,1,1,"N.Kurse"
   67  MENU 2,0,1,"Kaufen"
   68  MENU 2,1,1,"BMW"
   69  MENU 2,2,1,"Philips"
   70  MENU 2,3,1,"Kaufhof"
   71  MENU 2,4,1,"AEG"
   72  MENU 2,5,1,"D.B."
   73  MENU 2,6,1,"Fiat"
   74  MENU 2,7,1,"VW"
   75  MENU 2,8,1,"BASF"
   76  MENU 2,9,1,"Krupp"
   77  MENU 2,10,1,"Ford"
   78  MENU 3,0,1,"Verkaufen"
   79  MENU 3,1,1,"BMW"
   80  MENU 3,2,1,"Philips"
   81  MENU 3,3,1,"Kaufhof"
   82  MENU 3,4,1,"AEG"
   83  MENU 3,5,1,"D.B."
   84  MENU 3,6,1,"Fiat"
   85  MENU 3,7,1,"VW"
   86  MENU 3,8,1,"BASF"
   87  MENU 3,9,1,"Krupp"
   88  MENU 3,10,1,"Ford"
   89  MENU 4,0,1,"Tendenz"
   90  MENU 4,1,1,"Anzeigen"
   91  MENU 4,2,1,"+/-"
   92  MENU 5,0,1,"Ende"
   93  MENU 5,1,1,"    "
   94  Start:
   95  MENU ON
   96  ON MENU GOSUB Menue
   97  Ein:
   98  Xs=Xs+1
   99  IF Xs=30 THEN SOUND RESUME
```

```
  100  GOTO Start
  101  Berechne:
  102  IF yy=1 THEN WINDOW OUTPUT 5:COLOR 3,0:GOTO Nicht3
  103  FOR F=1 TO 10
  104  FOR i=1 TO 3:Pp(F,i)=INT(6*RND):NEXT
  105  NEXT F
  106  FOR F=1 TO 10
  107  IF Te$(F)="+" THEN Pa(F)=INT(12*RND)
  108  IF Te$(F)="-" THEN Pa(F)=INT(10*RND)-8
  109  NEXT F
  110  FOR F=1 TO 10
  111  WW(F)=We(F)+Pp(F,1)+Pp(F,2)-Pp(F,3)
  112  WW(F)=WW(F)+Pa(F)
  113  VE(F)=WW(F)-We(F)
  114  We(F)=WW(F)
  115  NEXT F
  116  Ko=Ko+(Ko*R/100)
  117  Ko=INT(Ko)
  118  AK=0:tt=K:tag=tag+1
  119  FOR i=1 TO 10
  120  zu=INT(3*RND)+1:AK=AK+an(i)*We(i)
  121  IF zu=1 THEN Te$(i)="+"
  122  IF zu<>1 THEN Te$(i)="-"
  123  NEXT i
  124  K=G+AK+Ko
  125  R=INT(10*RND)+1
  126  R=R/10
  127  GOSUB Musik
  128  yy=1
  129  GOSUB Aufbau
  130  GOTO Ein
  131  Kauf:
  132  E=MENU(1)
  133  WINDOW OUTPUT 5:COLOR 2,0
  134  PRINT:PRINT" Wieviel":PRINT:PRINT" kaufen":PRINT:INPU
T" "; aa:CLS
  135  WW=We(E):We(E)=We(E)*aa
  136  IF G<We(E) THEN WINDOW OUTPUT 5:We(E)=WW:COLOR 3,0:GO
TO Nicht1
  137  an(E)=an(E)+aa
  138  G=G-We(E):We(E)=WW:yy=0
  139  GOTO Aufbau
  140  Verkauf:
  141  E=MENU(1)
  142  WINDOW OUTPUT 5:COLOR 2,0
  143  PRINT:PRINT" Wieviel":PRINT:PRINT" verkaufen":PRINT:I
NPUT ";aa:CLS
  144  IF an(E)=0 OR an(E)<aa THEN WINDOW OUTPUT 5:COLOR 3,0
:GOTO Nicht2
  145  an(E)=an(E)-aa:WW=We(E):We(E)=We(E)*aa
  146  G=G+We(E):We(E)=WW:yy=0
  147  GOTO Aufbau
  148  Tendenz:
  149  IF MENU(1)=2 THEN Veraenderungen
  150  WINDOW OUTPUT 3
  151  RESTORE Datas
  152  CLS
  153  PRINT"  BMW   Philips  Kaufhof  AEG   D.B.   Fiat
";
  154  PRINT" VW   BASF   Krupp   Ford"
  155  COLOR 2,0:PRINT:PRINT"  ";
  156  FOR i=1 TO 10
  157  READ l
  158  PRINT Te$(i);
  159  PRINT SPC(l+2);
  160  NEXT i
  161  Taste:
  162  WINDOW OUTPUT 5:COLOR 3,0
  163  PRINT:PRINT" Bitte Taste":PRINT:PRINT"  Druecken"
  164  WHILE INKEY$=""
  165  WEND
  166  CLS
  167  GOTO Aufbau
  168  Veraenderungen:
  169  WINDOW OUTPUT 3
  170  RESTORE Datas
  171  CLS
  172  PRINT"  BMW   Philips  Kaufhof  AEG   D.B.   Fiat   "
;
  173  PRINT" VW   BASF   Krupp   Ford"
  174  COLOR 2,0:PRINT:PRINT" ";
  175  FOR i=1 TO 10
  176  READ l
  177  PRINT VE(i);
  178  PRINT SPC(l);
  179  NEXT i
  180  GOTO Taste
  181  GOTO Aufbau
  182  GOTO Start
  183  Aufbau:
  184  WINDOW OUTPUT 2
  185  CLS:COLOR 7,0
  186  PRINT:PRINT:PRINT" ";G;" DM"
  187  WINDOW OUTPUT 4
  188  CLS:COLOR 7,0
  189  PRINT:PRINT:PRINT" ";K;" DM"
  190  WINDOW OUTPUT 3
  191  CLS :COLOR 6,0
  192  PRINT"  BMW   Philips  Kaufhof  AEG   D.B.   Fiat   "
;
  193  PRINT" VW   BASF   Krupp   Ford"
  194  RESTORE Datas
  195  COLOR 4,0
  196  FOR i=1 TO 10
  197  READ l
  198  PRINT USING"###";an(i);
  199  PRINT SPC(l);
  200  NEXT i
  201  RESTORE Datas
  202  COLOR 5,0
  203  PRINT:PRINT" ";
  204  FOR u=1 TO 10
  205  READ l
  206  PRINT USING"###";We(u);
  207  PRINT SPC(l);
  208  NEXT u
  209  RETURN
  210  Datas:
  211  DATA 5,6,5,3,5,3,3,4,5,0
  212  Menue:
  213  MENU STOP
  214  ON MENU(0) GOSUB Berechne,Kauf,Verkauf,Tendenz,Ende
  215  MENU ON
  216  RETURN
  217  Ende:
  218  WINDOW OUTPUT 3:GOSUB Endmusik
  219  CLS:COLOR 2,0:PRINT:K=G+AK+Ko
  220  PRINT" Nach "tag" Jahren haben sie keine Lust mehr"
  221  IF K<KF THEN KF=KF-K:PRINT" Sie haben ";KF;" DM Verlu
st";:GOTO Ende1
  222  IF K>=KF THEN KF=K-KF:PRINT" Sie erwirtschafteten ";K
F" DM";
  223
  224  Ende1:
  225  KF=ABS(KF/tag):PRINT "  Das sind ";
  226  PRINT USING"#####.##";KF;:PRINT "DM pro Tag"
  227  MENU RESET:END
  228  Musik:
  229  FOR x=20 TO 3000 STEP 100
  230  SOUND x,1.1,40,0
  231  SOUND x,1.3,40,1
  232  SOUND x,1.6,40,2
  233  SOUND x,2,40,3
  234  NEXT x
  235  RETURN
  236  Endmusik:
  237  FOR x=3000 TO 20 STEP -100
  238  SOUND x,1.1,40,0:SOUND x,1.3,40,1:SOUND x,1.6,40,2:SO
UND x,2,40,3:NEXT
  239  FOR x=3000 TO 20 STEP -100
  240  SOUND x,2,40,3:SOUND x,1.6,40,2:SOUND x,1.3,40,1:SOUN
D x,1.1,40,0:NEXT
  241  RETURN
  242  Nicht1:
  243  PRINT:PRINT" So viel Geld":PRINT:PRINT" haben Sie ":P
RINT:PRINT" nicht !!"
  244  GOTO Schleife
  245  Nicht2:
  246  PRINT:PRINT" So viele":PRINT:PRINT" Aktien haben":PRI
NT:PRINT" sie nicht !"
  247  GOTO Schleife
  248  Nicht3:
  249  PRINT:PRINT" Sie muessen":PRINT:PRINT" kaufen oder":P
RINT:PRINT" verkaufen !!"
  250  Schleife:
  251  FOR i=1 TO 7000:NEXT i:CLS:GOTO Start
```

# C64 Listings

## Unheimliche Begegnung

Obgleich das Programm den gleichen Titel hat, wie der bekannte Spielberg-Film, so ist die Handlung doch völlig anders. Während der Filmheld erst Kontakte mit Außerirdischen knüpft, ist der Held dieses Programm bereits von Venusiern auf dem Planeten gefangen worden und in einem Gefängnisturm eingesperrt. Ihre Aufgabe ist es, die Rolle des Helden zu übernehmen und zu fliehen. Im ersten Teil des Programms müssen Sie den Ausgang aus dem dreistöckigen Gefängnis finden. Sie bewegen sich in dieser Adventure-Sequence auf drei Ebenen, die durch die Farben Gelb '1. Stock), Rot (zweiter Stock) und Grün (Erdgeschoß) gekennzeichnet sind. Auf Ihrem Weg in die Freiheit können Sie verschiedene Gegenstände finden und mitnehmen. Wechsel zwischen den Ebenen erfolgen durch Fahrstühle oder einen Transmitter. Damit es nicht zu leicht wird, befinden sich in manchen Räumen Fallen und Wachtposten.

Das Adventure benötigt deutschsprachige Eingaben und wird von guten grafischen Effekten begleitet. Haben Sie den Ausgang des Turmes gefunden, so geht es in die zweite Spielphase. In dieser Spielsequence müssen Sie mit einer Raumfähre den Planeten verlassen, den Asteroidengürtel passieren und die Erde erreichen. In diesem Teil erfordert das Steuern der Raumfähre gute Reaktionen, da der Flug eine äußerst schnelle Angelegenheit ist.

## Disks of Tron

Nach dem Starten des Programms mit Run wird zuerst der gesamte Zeichensatz neu definiert. Dieser Vorgang dauert ca. 20 Sekunden.

Danach erscheint die Titelseite des Programms und nach Drükken einer Taste befindet man sich im Spielmodus. Der Pfeil, der im unteren Bildschirmrand steht, ist auf den Befehl Game-Play gerichtet. Wird nun der Feuerknopf betätigt, so kann das Spiel beginnen. Drückt man aber zuerst den Joystick nach rechts, so blinkt der Pfeil mit dem Befehl Instruction Set und nach Betätigen des Feuerknopfes erscheint die Spielanleitung auf dem Bildschirm.

Zum Spielablauf:

Als Spielfigur 'Tron' befinden Sie sich auf der vorderen Scheibe und Ihr Gegner 'Zark' auf der hinteren. Ihre Aufgabe ist es, Zark so oft wie nur möglich abzuschießen. Dazu haben Sie eine kleine Scheibe, die Disk, zur Verfügung.Nach Drücken des Feuerknopfes bewegt sich die Scheibe auf das Zielkreuz zu, welches sich im Hintergrund des Bildschirmes befindet. Das Kreuz kann mit dem Joystick nach links und nach rechts bewegt werden. Auf die gleiche Weise ist es möglich, die Spielfigur Tron zu bewegen.

**Die beiden Listings befinden sich hintereinander im Heft. Erst Unheimliche Begegnung, danach Discs of Tron**

**Beachten Sie bitte die Eintipp-Hinweise auf Seite 79.**

```
 0 rem[SPACE][2][0][0]*
 1 rem[SPACE]*[0][0][SPACE]adventura[0][0][SPAC
E]*
 2 rem[SPACE]*[0][0][SPACE]1984[SPACE]by[0][0]
SPACE]*
 3 rem[SPACE]*[SPACE]martin[SPACE]kie[0][0]s
we[0][0]ter[SPACE]*
 4 rem[SPACE][2][0][0]*
 6 print"[ush CLR][2][0][0][DOWN]"tab(8)"[ct2]b
i[0][0]te[SPACE]warten[SPACE]([0][0]1[SPACE]sek
.)"chr$(8):ti$="[0][0]0"
 7 rem
 8 rem[SPACE]aLphabeth
 9 rem
 10 data126,129,189,161,161,189,129,126
 15 data[0][0] 2,[0][0] 6,[0][0] 0, 30, 54,102,198,
198
 20 data248,252,[0][0] 0,252,252,[0][0] 4,252,24
8
 25 data124,254,[0][0] 0,192,192,192,254,124
 30 data252,254,[0][0] 0,198,198,198,254,252
 35 data254,254,[0][0] 0,[0][0]24,[0][0]24,128,254,2
54
 40 data254,254,[0][0] 0,[0][0]24,[0][0]24,128,128,1
28
 45 data124,254,[0][0] 0,158,130,130,254,124
 50 data198,198,[0][0] 0,254,254,198,198,19
8
 55 data 60, 60,[0][0] 0, 24, 24, 24, 60, 60
 60 data254,254,[0][0] 0,[0][0] 6,[0][0] 6,198,252,1
20
 65 data198,204,[0][0] 0,240,240,216,204,19
8
 70 data192,192,[0][0] 0,192,192,192,254,254
 75 data130,198,[0][0] 0,254,214,198,198,198
 80 data134,198,[0][0] 0,246,[0][0]2,206,198,19
4
 85 data124,254,[0][0] 0,198,198,198,254,124
 90 data252,254,[0][0] 0,254,252,192,192,19
2
 95 data124,254,[0][0] 0,198,214,206,254,124
 100 data252,254,[0][0] 0,254,252,240,216,20
4
 105 data126,254,[0][0] 0,252,126,[0][0] 2,254,
252
 110 data254,254,[0][0] 0, 16, 16, 16, 16, 1
6
 115 data198,198,[0][0] 0,198,198,198,254,12
4
 120 data198,198,[0][0] 0,198, 68,108, 56, 1
6
 125 data198,198,[0][0] 0,198,214,214, 84, 4
0
 130 data130,198,[0][0] 0, 56, 56,198,198,13
0
 135 data130,198,[0][0] 0, 56, 56, 16, 16, 1
6
 140 data254,254,[0][0] 0,126, 48, 96,254,25
4
 150 fort=0to215:reada:poke573[0][0]4+t,a:pr
int"[HOME]"ti$:next
 157 rem
 158 rem[SPACE]zahLen
 159 rem
 160 data124,254,[0][0] 0,198,246,230,254,12
4
 165 data 14, 30,[0][0] 0,102,[0][0] 6,[0][0] 6,[0][0] 6
,[0][0] 6
 170 data124,254,[0][0] 0, 24, 48, 96,254,2
54
 175 data124,254,[0][0] 0, 62, 62,[0][0] 2,254,1
24
 180 data192,192,[0][0] 0,216,254, 24, 24,
24
 185 data254,254,[0][0] 0,252,254,[0][0] 2,254,1
24
 190 data124,254,[0][0] 0,252,254,194,254,12
4
 195 data254,254,[0][0] 0, 12, 24, 48, 96,19
2
 200 data124,254,[0][0] 0,124, 68,130,254,12
4
 205 data124,254,[0][0] 0,254,126,[0][0] 6,254,1
24
 210 fort=0to79:reada:poke5[0][0]728+t,a:pri
nt"[HOME]"ti$:next
 217 rem
 218 rem[SPACE]diverses
 219 rem
 220 data[0][0] 0,[0][0] 0,[0][0] 0,[0][0] 0,[0][0] 0,[0][0] 0,[0]
[0] 0,[0][0] 0
 225 data 24, 24, 24, 24, 24,[0][0] 0, 24, 2
4
 230 data[0][0] 0,[0][0] 0,[0][0] 0,[0][0] 0, 24, 24, 48
, 64
 235 data[0][0] 0,[0][0] 0,[0][0] 0,126,126,[0][0] 0,[0][0]
0,[0][0] 0
 240 data[0][0] 0,[0][0] 0,[0][0] 0,[0][0] 0,[0][0] 0, 24, 2
4, 24
 245 data[0][0] 0, 24, 24,[0][0] 0,[0][0] 0, 24, 24
,[0][0] 0
```

```
 290 fort=0to15:reada:poke576[][]0+t,a:nex
t
 295 fort=0to23:reada:poke57696+t,a:next
 300 fort=0to7:reada:poke57808+t,a:next
 305 print"[HOME]"ti$
 309 rem[SPACE]fragezeichen
 310 data126,192,3,6,12,24,0,24
 317 rem
 318 rem[SPACE]kaestchen[SPACE]w[][]agrec
ht
 319 rem
 320 data 16, 16, 16,2[][]5, 16, 16, 16, 1
6
 321 data 16, 16, 16, 16,2[][]5, 16, 16,
16
 322 data 16, 16, 16, 16, 16,2[][]5, 16, 1
6
 323 data 16, 16, 16, 16, 16, 16,2[][]5, 1
6
 324 data 16, 16, 16, 16, 16, 16, 16,2[][]
5
 325 data2[][]5, 16, 16, 16, 16, 16, 16, 1
6
 326 data 16,2[][]5, 16, 16, 16, 16, 16, 1
6
 327 data 16, 16,2[][]5, 16, 16, 16, 16, 1
6
 328 rem
 329 rem[SPACE]kaestchen[SPACE]senkrecht
 330 data 16, 16, 16,2[][]5, 16, 16, 16, 1
6
 331 data 32, 32, 32,2[][]5, 32, 32, 32, 3
2
 332 data 64, 64, 64,2[][]5, 64, 64, 64, 6
4
 333 data128,128,128,2[][]5,128,128,128,12
8
 334 data[][] 1,[][] 1,[][] 1,2[][]5,[][] 1,[][] 1,[]
[] 1,[][] 1
 335 data[][] 2,[][] 2,[][] 2,2[][]5,[][] 2,[][] 2,[]
[] 2,[][] 2
 336 data[][] 4,[][] 4,[][] 4,2[][]5,[][] 4,[][] 4,[]
[] 4,[][] 4
 337 data[][] 8,[][] 8,[][] 8,2[][]5,[][] 8,[][] 8,[]
[] 8,[][] 8
 350 fort=0to135:reada:poke57848+t,a:pri
nt"[HOME]"ti$:next
 355 rem[SPACE]cursor
 356 data0,0,0,0,0,0,2[][]5,2[][]5
 357 fort=0to7:reada:poke58624+t,a:next
 358 rem
 359 rem[SPACE]symboLe
 360 data[][] 0,127, 64, 95, 87, 91, 93, 9
4
 361 data 94, 93, 91, 87, 95, 64,127,[][]
0
 362 data[][] 0,254,[][] 2,250,234,218,186,1
[][]2
 363 data1[][]2,186,218,234,250,[][] 2,254,[]
[] 0
 364 data[][] 0,[][] 0,[][] 0,2[][]5,254,178, 16
,[][] 0
 365 data 28, 62,[][]19,235,[][]19, 62, 28,[]
[] 0
 366 data[][] 0,126,129,184,184,2[][]5,126,
60
 367 data 86,149,181,181,181,181,2[][]5,2[]
[]5
 368 data 24, 24, 24, 60,126,2[][]5,2[][]5,1
26
 369 data[][] 0,127, 63, 63, 56,[][]12,[][]24,
[][]24
 370 data[][] 0,2[][]5,2[][]5,[][] 0,[][] 0,[][] 0,[]
[] 0,[][] 0
 399 fort=0to87:reada:poke57984+t,a:prin
t"[HOME]"ti$:next
 400 poke53272,24:poke56576,148:poke648,
196
 410 gosub4[][]0
 420 rem
 421 rem[SPACE]sprites
 422 rem
 423 rem[SPACE]raumanzug
 425 data0,0,0,0,0,0,3,192,0,15,240,0,63
,252,0,[][]1,246,0,[][]1,246,0,[][]1,246,238
 430 data[][]1,246,238,[][]1,246,198,[][]1,246
,198,15,246,198,15,240,0,14,[][]12,0,14,[][]
12
 435 data96,14,[][]12,240,14,[][]13,248,14,[]
[]13,248,14,[][]13,248,14,[][]12,0,0,0,0,0
 436 rem[SPACE]wache
 440 data0,3,0,0,7,128,0,7,128,0,3,0,0,7
,128,0,15,192,32,2[][]5,192,127,2[][]5,192
 441 data12,127,192,7,15,192,0,15,192,0,
7,128,0,15,128,0,31,128,0,57,128,0,241
 442 data128,0,[][]25,128,0,193,128,0,193,
136,0,193,252,3,193,242,0
 449 rem[SPACE]teLeporter
 450 data0,0,0,0,2[][]5,0,3,1[][]9,192,15,23
9,240,31,239,248,63,2[][]5,252,123,0,62,[][]
12
 451 data0,14,96,2,6,64,2,2,64,2,2,32,12
4,4,16,68,8,8,124,16,4,68,32,2,68,64
 452 data2,68,64,3,2[][]5,192,7,2[][]5,[][]24,
31,2[][]5,248,127,2[][]5,254
 455 fort=0to190:reada:poke49536+t,a:nex
t
 465 vf=53248:poke5[][]192,6:poke5[][]193,7:
pokevf+39,1:pokevf+40,0
 466 poke5[][]194,8:pokevf+41,1:pokevf+4,[]
[]10:pokevf+5,130
 470 pokevf,[][]10:pokevf+1,130:pokevf+2,[]
[]10:pokevf+3,130
 500 rem
 501 rem[SPACE]maschinenprogra[][]me
 502 rem
 503 rem[][][SPACE]scro[][]Len
 510 data162, 18,173,[][] 0,192,160, 18,14
5
 515 data174,136,208,251, 24,165,174,105
 520 data 40,1[][]3,174,165,175,105,[][] 0,1
[][]3
 525 data175,202,208,230,238,[][] 0,192,17
3
 530 data[][] 0,192,201, 72,208,[][] 5,169,
64
 535 data141,[][] 0,192,169,121,1[][]3,174,1
69
 540 data196,1[][]3,175,96
 543 rem
 544 rem[SPACE]einfaerben
 545 rem
 550 data162, 18,169,[][] 0,160, 18,145,17
0
 555 data136,208,251, 24,165,170,105, 40
 560 data1[][]3,170,165,171,105,[][] 0,1[][]3,
171
 565 data202,208,231, 96
 600 fort=49154to49154+79:reada:poket,a:
s=s+a:next
 601 ifs<>10784thenprint"fehLer[SPACE]in
[SPACE]maschinenprogra[][]men"
 610 poke174,121:poke175,196:poke49153,7
2:poke49152,64
 620 poke170,128:poke171,216
 700 rem
 701 rem[SPACE]var.dekLiration
 702 rem
 703 rem[SPACE]data[SPACE]pL(x,y,z)
 704 data0,1,2,0,4,0,0,6,7,1,0,2,3,4,5,0
,6,7,1,2,0,3,4,5,0,6,7
 705 forz=2to0step-1:fory=0to2:forx=0to2
:reada:pl(x,y,z)=a:nextx:nexty:nextz
 710 x=1:y=1:z=1:z$(0)="[ct6]gruen":z$(1
)="[ct8]geLb[SPACE]":z$(2)="[c=3]rot[][]
SPACE]":a=[][]2
 720 v=5
 730 datasaeure,laser,teleporter,lampe,r
aumanzug,schlue[][]sel
 740 fort=0to5:readmo$(t):mi$(t)=mo$(t):
next
 744 data3,5,6,7,8,9
 745 fort=0to5:readpq(t):next
 900 print"[shCLR]":poke49209,7:gosub204
5
 910 sys49154:poke53281,[][]1
```

# C64 Listings

```
 997 rem
 998 rem■■[SPACE]hauptprogra■■m
 999 rem
 1000 print"[HOME][DOWN][RIGHT][ct2]sie[
SPACE]sind[SPACE]im[SPACE]sektor:[SPACE]
";z$(z); pl(x,y,z)
 1008 print"[HOME]■■[DOWN]"tab(21)"[■■
SPACE]":pq(2)=6
 1010 goto2050
 1012 ge(2)=0
 1015 onogoto24■■0,2405,2410,2415,2420,2
425,2430,2435,2■■40
 1020 o=0:print"[HOME]■■[DOWN]"tab(21)a
$
 1022 f$="■■[SPACE]"
 1025 goto25■■0
 1026 goto27■■0
 1027 print"[HOME]■■[DOWN][RIGHT][ct1]"
f$
 1030 input"[HOME]■■■[DOWN][ct2][SPACE]k
omando";k$
 1040 print"[shUP]"tab(8)"■■[SPACE]":po
kevf+21,0
 1045 print"■■[SPACE]"
 1048 print"[HOME]■■[DOWN]■■[SPACE]"
 1050 goto■■2■■0
 1060 onvgosub2010,2020,2030,2040,2043
 1070 mi$(2)="teLeporter":v=4:a$="[■■[SP
ACE]"
 1080 goto1■■0
 2000 rem
 2001 rem[SPACE]unte■■routinen
 2002 rem
 2003 rem[SPACE]o[SPACE]u
 2010 poke49157,0:poke49183,0:poke49186,
0:poke49189,72:poke49193,64:poke49195,0
 2011 poke49152,64:poke49182,238
 2012 fort=1to56:sys49154:forr=1to20:nex
tr:next:return
 2015 rem[SPACE]u[SPACE]o
 2020 poke49157,0:poke49183,0:poke49186,
0:poke49189,64:poke49193,72:poke49195,0
 2021 poke49152,71:poke49182,206
 2022 fort=1to56:sys49154:forr=1to20:nex
tr:next:return
 2025 rem[SPACE]L[SPACE]r
 2030 poke49157,1:poke49183,1:poke49186,
1:poke49189,80:poke49193,72:poke49195,1
 2031 poke49153,72:poke49182,238
 2032 fort=1to56:sys49154:forr=1to20:nex
tr:next:return
 2035 rem[SPACE]r[SPACE]L
 2040 poke49157,1:poke49183,1:poke49186,
1:poke49189,71:poke49193,79:poke49195,1
 2041 poke49153,79:poke49182,206
 2042 fort=1to56:sys49154:forr=1to20:nex
tr:next
 2043 return
 2044 rem[SPACE]einfaerben
 2045 poke170,121:poke171,216:sys49206:r
eturn
 2050 ifz=1then21■■0
 2060 ifz=2then2150
 2070 ifx=0andy=0thena$="fahrstuhL■■[SPA
CE]":o=1
 2071 ifx=1andy=0thena$="wache[■■[SPACE]
":o=2
 2072 ifx=0andy=1thena$=mi$(0):o=pq(0)
 2073 ifx=1andy=2thena$="fa■■Lgrube■■[SP
ACE]":o=4
 2074 ifx=2andy=1orx=2andy=2thenw=1
 2075 ifx=2andy=2thena$=mi$(5):o=pq(5)
 2080 goto1012
 2100 ifx=0andy=0orx=1andy=2thena$="fahr
stuhL■■[SPACE]":o=1
 2105 ifx=1andy=1thena$="start[■■[SPACE]
"
 2110 ifx=2andy=0thena$=mi$(1):o=pq(1)
 2120 goto1012
 2150 ifx=1andy=0thena$="teLeporter■■[S
PACE]":o=pq(2)
 2151 ifx=2andy=0thena$=mi$(3):o=pq(3)
 2152 ifx=1andy=1thena$=mi$(4):o=pq(4)
 2153 ifx=1andy=2thena$="fahrstuhL■■[SPA
CE]":o=1
 2154 ifx=2andy=2thena$="keinL[SPACE]saue
rsto■■f"
 2155 ifx=0andy=1orx=0andy=2thenw=2
 2160 goto1012
 2200 rem
 2201 rem[SPACE]komandoauswertung
 2202 rem
 2205 q=pl(x,y,z):q1=x:q2=y:q3=z
 2210 ifk$="n"andy>0theny=y-1:v=1:goto23
■■0
 2211 ifk$="o"andx<2thenx=x+1:v=3:goto23
■■0
 2212 ifk$="s"andy<2theny=y+1:v=2:goto23
■■0
 2213 ifk$="w"andx>0thenx=x-1:v=4:goto23
■■0
 2220 ifk$="h"andz=0andq=1thenz=z+1:v=5:
poke49209,7:gosub2045:goto23■■0
 2221 ifk$="h"andz=1andq=6thenz=z+1:v=5:
poke49209,10:gosub2045:goto23■■0
 2222 ifk$="r"andz=2andq=6thenz=z-1:v=5:
poke49209,7:gosub2045:goto23■■0
 2223 ifk$="r"andz=1andq=1thenz=z-1:v=5:
poke49209,5:gosub2045:goto23■■0
 2299 v=5
 2300 ifpl(x,y,z)=0thenx=q1:y=q2:z=q3:v=
5
 2310 ifleft$(k$,4)="ni■■m"then2325
 2312 ifleft$(k$,4)="List"then2350
 2314 ifleft$(k$,7)="benutze"then2365
 2320 goto1060
 2325 fort=0to5
 2326 ifk$=left$(k$,5)+mo$(t)thenmi$(t)=
"[■■[SPACE]":pq(t)=0:ge(t)=1
 2327 next:sys49154:goto1060
 2330 fort=0to5
 2349 rem[SPACE]List
 2350 print"[HOME]■■[DOWN]"
 2352 fort=0to5
 2354 ifge(t)=1thenprinttab(24)mo$(t)
 2356 next
 2358 fort=1to5■■0:next:print"[HOME]■■[
DOWN]":fort=0to5:printtab(24)"[■■[SPACE]
":next
 2360 goto1060
 2365 ifz=2andx=0andy=1orz=2andx=0andy=2
then29■■0
 2368 ifk$=left$(k$,8)+mo$(2)andz=2andx=
1andy=0thengoto3■■0
 2375 goto1060
 2397 rem
 2398 rem[SPACE]symboLe[SPACE]zeichnen
 2399 rem
 2400 gosub24■■9
 2401 printtab(10)"[shp][shr]":printtab
(10)"[shq][shs]"
 2402 goto1020
 2405 pokevf+21,2
 2406 goto1020
 2410 gosub24■■9
 2411 printtab(10)"■■[SPACE]":printtab(1
0)"[SPACE][shx][SPACE]":printtab(10)"■■[
SPACE]":goto1020
 2415 gosub24■■9
 2416 fort=1to4:printtab(10)"■■[SPACE]":
next:goto1020
 2420 gosub24■■9
 2421 printtab(10)"[shy][shz]":goto1020
 2425 pokevf+21,4
 2426 goto1020
 2430 gosub24■■9
 2431 printtab(10)"[shv]":printtab(10)"[
shw]":goto1020
 2435 pokevf+21,1
 2436 goto1020
 2440 gosub24■■9
 2441 printtab(10)"[sht][shu]":goto1020
 2499 print"[HOME]■■[DOWN]":return
 2500 rem
 2501 rem[SPACE]weLche[SPACE]richtung
 2502 rem
```

```
 2505 print"{HOME}{5 RVS}{DOWN}"tab(23)"{ct
1}richtung:":print"{DOWN}"tab(25)"{8 RVS}{SPA
CE}{shUP}"
 2510 ify<2andpl(x,y+1,z)<>0thenprinttab
(26)"{c=6}s{shUP}"
 2520 ifx<2andpl(x+1,y,z)<>0thenprinttab
(27)"{c=6}o{shUP}"
 2522 ify=0then2525
 2524 ify>0andpl(x,y-1,z)<>0thenprinttab
(25)"{c=6}n{shUP}"
 2525 ifx=0then2530
 2526 ifx>0andpl(x-1,y,z)<>0thenprinttab
(28)"{c=6}w{shUP}"
 2530 ifz=0andpl(x,y,z)=1orz=1andpl(x,y,
z)=6thenprinttab(29)"{c=6}h{shUP}"
 2535 ifz=2andpl(x,y,z)=6orz=1andpl(x,y,
z)=1thenprinttab(30)"{c=6}r{shUP}"
 2550 goto1026
 2700 rem
 2701 rem{SPACE}reaktionen
 2702 rem
 2710 ifge(0)=1thenl$="die{SPACE}saeure{
SPACE}ist{SPACE}ausgeLaufen{SPACE}und{S
PACE}hat{SPACE}sie{SPACE}veraetzt.":got
o5{2 RVS}0
 2720 ifz=0andx=1andy=2thenl$="sie{SPACE
}sind{SPACE}in{SPACE}die{SPACE}fa{2 RVS}Lgru
be{SPACE}gefa{2 RVS}Len":goto5{2 RVS}0
 2725 ifz=0andx=1andy=0thenl$="die{SPACE
}wache{SPACE}hat{SPACE}sie{SPACE}eLimini
ert":goto5{2 RVS}0
 2740 ifz=0andx=2andy=1andge(3)=0thengot
o28{2 RVS}0
 2750 ifz=2andx=2andy=2andge(4)=0thenl$=
"sie{SPACE}sind{SPACE}erstickt":goto5{2 RVS}0
 2760 ifge(5)=1thenpl(0,1,2)=3:pl(0,2,2)
=5
 2770 ifz=2andx=0andy=1orz=2andx=0andy=2
thena$="verschLo{2 RVS}senes{SPACE}tor"
 2780 ifz=2andx=0andy=2orz=2andx=0andy=1
thenf$="sie{SPACE}stehen{SPACE}vor{SPACE
}einem{SPACE}tor"
 2790 goto1027
 2800 rem{SPACE}dunkeL{SPACE}werden
 2805 print"{HOME}{2 RVS}{DOWN}{RIGHT}{ct2}
es{SPACE}wird{SPACE}dunkeL"
 2810 poke53281,0
 2811 poke49209,15:gosub2045
 2812 fort=1to4{2 RVS}0:next
 2813 poke49209,12:gosub2045
 2814 fort=1to4{2 RVS}0:next
 2815 poke49209,{2 RVS}1:fort=1to4{2 RVS}0:next
 2816 poke49209,0:gosub2045:fort=1to4{2 RVS}0
:next:print"{shCLR}"
 2817 l$="sie{SPACE}sind{SPACE}gestuertz
t{SPACE}und{SPACE}haben{SPACE}sich{SPACE
}das{2 RVS}{SPACE}genick{SPACE}gebrochen"
 2820 goto5{2 RVS}0
 2900 ifk$=left$(k$,8)+mo$(5)andge(5)=1t
hen2910
 2905 goto1060
 2910 print"{HOME}{2 RVS}{DOWN}{RIGHT}{ct2}d
as{SPACE}tor{SPACE}zur{SPACE}freiheit{SP
ACE}oe{2 RVS}fnet{SPACE}sich"
 2915 goto6{2 RVS}0
 3000 rem{SPACE}teLeporter{SPACE}beamt
 3001 print"{HOME}{DOWN}{ct1}{2 RVS}{SPACE}te
Leporter{2 RVS}{SPACE}beamt{2 RVS}{SPACE}sie{2 RVS}{SPA
CE}{2 RVS}!"
 3005 x=int(rnd(1)*2)
 3006 y=int(rnd(1)*2)
 3007 z=int(rnd(1)*2)
 3008 ifpl(x,y,z)=0then3{2 RVS}05
 3009 ifz=0thenpoke49209,5
 3010 ifz=1thenpoke49209,7
 3011 ifz=2thenpoke49209,10
 3012 gosub2045
 3013 a$="{3 RVS}{SPACE}"
 3014 goto1{2 RVS}0
 4000 rem
 4001 rem{SPACE}anfangsbiLd
 4002 rem
 4005 print"{shCLR}{8 RVS}{DOWN}{ct3}":poke53
281,0:poke53280,0
 4006 print"{3 RVS}{SPACE}{4 RVS}{sh*}"
 4007 print"{3 RVS}{SPACE}{4 RVS}{sh*}"
 4008 print"{3 RVS}{SPACE}{5 RVS}{sh*}{ct8}unheim
Liche{ct3}{3 RVS}{sh*}"
 4009 print"{3 RVS}{SPACE}{4 RVS}{sh*}"
 4010 print"{3 RVS}{SPACE}{5 RVS}{sh*}{ct8}begeg
nung{ct3}{5 RVS}{sh*}"
 4011 print"{3 RVS}{SPACE}{4 RVS}{sh*}"
 4012 print"{3 RVS}{SPACE}{5 RVS}{sh*}{ct8}der{ct
3}{sh*}{c=6}3.art{ct3}{5 RVS}{sh*}"
 4013 print"{3 RVS}{SPACE}{4 RVS}{sh*}"
 4014 print"{3 RVS}{SPACE}{4 RVS}{sh*}"
 4015 print"{8 RVS}{DOWN}{ct2}{5 RVS}{SPACE}{@}{SPA
CE}1984{SPACE}su{2 RVS}ny{SPACE}soft"
 4016 print"{DOWN}{5 RVS}{SPACE}wri{2 RVS}ten{SPAC
E}by{2 RVS}{SPACE}martin{SPACE}kie{2 RVS}swe{2 RVS}ter
"
 4017 print"{DOWN}{ct6}{4 RVS}{SPACE}pre{2 RVS}s{
SPACE}any{SPACE}key"
 4018 poke198,0:wait198,1
 4019 print"{shCLR}{DOWN}{RIGHT}{ct3}die
{SPACE}erde{SPACE}hat{SPACE}erstmaLs"
 4020 print"{DOWN}{RIGHT}{ct8}kontakt{S
PACE}zu{SPACE}au{2 RVS}ser{SPACE}-"
 4021 print"{DOWN}{RIGHT}{ct3}irdischen{
SPACE}wesen."
 4022 print"{DOWN}{RIGHT}{ct8}sie{SPACE}
sind{SPACE}Leiter{SPACE}der"
 4023 print"{DOWN}{RIGHT}{ct3}begrue{2 RVS}su
ngsgru{2 RVS}pe."
 4024 print"{DOWN}{RIGHT}{ct8}die{SPACE}
gesamte{SPACE}gru{2 RVS}pe"
 4025 print"{DOWN}{RIGHT}{ct3}wird{SPACE
}jedoch{SPACE}gewaLt-"
 4026 print"{DOWN}{RIGHT}{ct8}sam{SPACE}
entfuehrt."
 4027 print"{DOWN}{RIGHT}{ct3}sie{SPACE}
erwachen{SPACE}in{SPACE}der"
 4028 print"{DOWN}{RIGHT}{ct8}fremdartig
en{SPACE}weLt"
 4029 print"{DOWN}{RIGHT}{ct3}der{SPACE}
au{2 RVS}serirdischen."
 4030 print"{DOWN}{ct6}{8 RVS}{SPACE}pre{2 RVS}s{
SPACE}any{SPACE}key"
 4031 poke198,0:wait198,1
 4032 print"{shCLR}{DOWN}{RIGHT}{ct8}be
fehLe:"
 4033 print"{DOWN}{RIGHT}{ct8}n{ct3}{5 RVS}.
nord"
 4034 print"{DOWN}{RIGHT}{ct8}o{ct3}{5 RVS}.o
st"
 4035 print"{DOWN}{RIGHT}{ct8}s{ct3}{5 RVS}.s
ued"
 4036 print"{DOWN}{RIGHT}{ct8}w{ct3}{5 RVS}.
west"
 4037 print"{DOWN}{RIGHT}{ct8}h{ct3}{5 RVS}.h
och"
 4038 print"{DOWN}{RIGHT}{ct8}r{ct3}{5 RVS}.
runter"
 4039 print"{DOWN}{RIGHT}{ct8}benutze{S
PACE}{ct3}{5 RVS}."
 4040 print"{DOWN}{RIGHT}{ct8}ni{2 RVS}m{SPA
CE}{ct3}{5 RVS}."
 4041 print"{DOWN}{RIGHT}{ct8}List{ct3}{5 R
VS}.was{SPACE}man{SPACE}bei{SPACE}sich{SPA
CE}hat"
 4042 print"{8 RVS}{DOWN}{2 RVS}{RIGHT}{ct6}pre{2 R
VS}s{SPACE}any{SPACE}key"
 4043 poke198,0:wait198,1
 4044 print"{shCLR}":return
 5000 rem
 5001 rem{SPACE}spieLende{SPACE}negativ
 5002 rem
 5010 print"{HOME}{2 RVS}{DOWN}{RIGHT}{ct2}
"l$
 5020 print"{DOWN}{2 RVS}{RIGHT}{ct2}pre{2 RVS}s{S
PACE}space";
 5030 getr$:ifr$<>"{SPACE}"then5030
 5035 poke53281,5:poke53280,13:pokevf+21
,0
 5040 input"{shCLR}{8 RVS}{DOWN}{2 RVS}{RIGHT}{ct
1}neues{SPACE}spieL{SPACE}";i$
```

# C64 Listings

```
 5050 ifi$="ja"thenpoke63,169:poke64,1:p
oke65,216:poke■■6,20:poke53280,0:goto425
 5060 ifi$="nein"thenprint"■■{DOWN}{RIGH
T}druecke■■{SPACE}runstop:restore":poke6
48,4:end
 5070 goto5040
 6000 rem
 6001 rem{SPACE}spieLende{SPACE}positiv
 6002 rem
 6005 poke53281,0:poke53280,0:print"{shC
LR}{ct2}"
 6010 print"■■{DOWN}{RIGHT}sie{SPACE}hab
en{SPACE}es{SPACE}gescha■■ft{SPACE},"
 6012 print"■■{DOWN}{RIGHT}aus{SPACE}der
{SPACE}au■■serirdischen{SPACE}"
 6014 print"■■{DOWN}{RIGHT}weLt{SPACE}zu
{SPACE}fLiehen{SPACE}."
 6016 print"■■{DOWN}{RIGHT}sie{SPACE}mue
■■sen{SPACE}nun{SPACE}noch{SPACE}mit"
 6018 print"■■{DOWN}{RIGHT}einem{SPACE}
raumschi■■f{SPACE}zur{SPACE}"
 6020 print"■■{DOWN}{RIGHT}erde{SPACE}zu
ruec■■kehren{SPACE}."
 6022 print"{DOWN}{ct3}■■{RIGHT}sind{SP
ACE}sie{SPACE}bereit{SPACE}?"
 6024 print"{DOWN}■■{RIGHT}pre■■s{SPACE}
space"
 6026 print"{HOME}{■■}{DOWN}"tab(28)"{ct8
}z■■.Links"
 6028 print"■■{DOWN}"tab(28)"{ct8}?■■.re
chts"
 6050 geta$:ifa$<>"{SPACE}"then6050
 6055 ifge(4)=0thenl$="{shCLR}sie{SPACE}
haben{SPACE}keinen{SPACE}raumanzug,sie{
SPACE}sind{SPACE}er-{SPACE}stickt":goto5
■■0
 6060 ifge(1)=0thenl$="{shCLR}kein{SPACE
}Laser{SPACE}um{SPACE}das{SPACE}schLo■■s
{SPACE}aufzuschie■■sen":goto5■■0
 6066 print"{shCLR}■■■{DOWN}■■{RIGHT}{ct
8}sie{SPACE}fLiegen{SPACE}durch{SPACE}de
n{SPACE}zei■■tu■■neL{SPACE}!"
 6070 rem{SPACE}raumschi■■f
 6071 data0,0,0,0,0,0,0,0,0,1,2■■5,128,7
,2■■5,■■24,15,85,240,31,170,248,63,2■■5
 6072 data252,63,2■■5,252,1■■0,146,78,1■
■0,146,78,63,2■■5,252,15,2■■5,240,■■1,2■
■5,208
 6073 data16,0,8,■■12,0,14,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0
 6074 fort=0to63:reada:poke49536+t,a:nex
t
 6075 x=1■■5:pokevf,x:pokevf+1,80:pokevf
+39,3:print"{shCLR}{ct2}":pokevf+31,0
 6080 z=15:pokevf+21,1:y=14:ti$="■■0"
 6097 rem
 6098 rem■■{SPACE}tu■■neL
 6099 rem
 6100 print"{HOME}■■■{DOWN}■■■."
 6103 ifti$="■■030"then6250
 6105 ifpeek(197)=12thenx=x-3
 6106 ifpeek(197)=■■5thenx=x+3
 6107 pokevf,x
 6108 ifpeek(vf+31)=1then62■■0
 6110 z=z+int(rnd(1)*3)-1
 6111 ifz<1thenz=1
 6112 ifz>28thenz=28
 6120 print"{shUP}"tab(z)"■■{SPACE}"
 6140 ifpeek(197)=12thenx=x-4
 6150 ifpeek(197)=■■5thenx=x+4
 6160 pokevf,x
 6165 ifpeek(vf+31)=1then62■■0
 6170 pokevf+39,y
 6175 y=y-7:y=1-y:y=y+7
 6177 y=y+1
 6180 goto61■■0
 6200 l$="sie{SPACE}sind{SPACE}abgetuerz
t{SPACE}■■!"
 6201 goto5■■0
 6250 fort=1to15:print"{DOWN}":next
 6255 pokevf,2■■5:pokevf+1,50:pokevf+39,
7
 6260 print"{HOME}■■■{DOWN}{RIGHT}{ct3}si
e{SPACE}sind{SPACE}gere■■tet{SPACE}■■!"
 6265 print"■■{DOWN}{RIGHT}{ct2}die{SPAC
E}automatische{SPACE}"
 6270 print"■■{DOWN}{RIGHT}steuerung{SPA
CE}des{SPACE}raum-"
 6275 print"■■{DOWN}{RIGHT}schi■■fes{SPA
CE}bringt{SPACE}sie"
 6280 print"■■{DOWN}{RIGHT}zurueck{SPAC
E}zur{SPACE}erde."
 6281 print"{HOME}■■■{DOWN}{c=7}"
 6282 printtab(26)"■■■{SPACE}■■."
 6283 printtab(26)"■■{SPACE}■■."
 6284 printtab(26)"■■{SPACE}■■."
 6285 printtab(26)"■■{SPACE}■■."
 6286 printtab(26)"■■{SPACE}■■.{ct6}■■.{
c=7}"
 6287 printtab(26)"■■{SPACE}■■.{ct6}■■.{
c=7}"
 6288 printtab(26)"■■{SPACE}■■.{ct6}■■.{
c=7}"
 6289 printtab(26)"{SPACE}■■.{ct6}■■.{c=
7}"
 6290 printtab(26)"{SPACE}■■■.{ct6}■■.{
c=7}"
 6291 printtab(26)"■■■."
 6292 printtab(26)"■■■."
 6293 fort=50to250:pokevf+1,t
 6294 forr=1to30:nextr:next
 6295 fort=1to1■■0:next:goto5035■■
```

```
 5 poke53280,■■1:poke53281,■■1
 8 rem■■{SPACE}■■■{ARROWLEFT}
 10 rem{SPACE}~■■{SPACE}am■■c{SPACE}power{SPACE}c2
■■oLware{SPACE}(c)~
 20 rem{SPACE}~■■{SPACE}presents■■{SPACE}~
 30 rem{SPACE}~■■■{SPACE}~
 40 rem{SPACE}~■■{SPACE}discs{SPACE}of{SPACE}tron/
■■{SPACE}~
 45 rem{SPACE}~■■■{SPACE}~
 46 rem{SPACE}~■■■{SPACE}~
 50 rem{SPACE}~{SPACE}created{SPACE}by{SPACE}boc{S
PACE}and{SPACE}mert■■{SPACE}~
 55 rem{SPACE}~■■■{SPACE}~
 60 rem{SPACE}■■■{ARROWLEFT}
 100 rem{SPACE}data{SPACE}fuer{SPACE}maschinenrout
ine
 110 fori=832to832+■■3:reada:pokei,a:next
 115 data120,169,51,1■■3,1,169,0,1■■3,95,1■■3,90,
1■■3,■■8,169,208,1■■3,96,169,240
 120 data1■■3,89,169,■■24,1■■3,91,32,191,163,169,■
■5,1■■3,1,■■8,96
 125 sys832:poke850,160:sys832
 130 poke56576,peek(56576)and252
 135 poke53272,8:poke648,192
 140 print"{shCLR}{ct2}■■■{DOWN}■■■{SPACE}pLease{SP
ACE}wait{SPACE}a{SPACE}moment"
 145 print"■■{SPACE}whiLe{SPACE}the{SPACE}characte
rs{SPACE}are"
 146 print"■■■{SPACE}designed
 150 fori=0to1■■07:reada:poke573■■4+i,a:next
 155 rem{SPACE}data{SPACE}for{SPACE}writing
 160 data60,■■6,157,161,161,157,■■6,60
 162 data60,36,126,98,98,98,98,0
 164 data124,36,62,50,50,50,126,0
 166 data126,■■6,64,96,96,98,126,0
 168 data126,34,34,50,50,50,126,0
 170 data126,64,64,120,96,96,126,0
 172 data126,64,64,120,96,96,96,0
 174 data126,■■6,64,■■10,98,98,126,0
 176 data■■6,■■6,■■6,126,98,98,98,0
 178 data16,16,16,24,24,24,24,0
 180 data2,2,2,6,6,70,60,0
 182 data■■6,68,72,126,98,98,98,0
 184 data64,64,64,96,96,96,126,0
 186 data102,90,■■6,98,98,98,98,0
 188 data■■14,74,74,106,106,106,102,0
 190 data126,■■6,■■6,98,98,98,126,0
 192 data126,■■6,■■6,126,96,96,96,0
 194 data0,0,0,0,0,0,0,0
 196 data126,■■6,■■6,126,1■■0,98,98,0
 198 data126,■■6,64,126,6,70,126,0
```

```
200 data124,16,16,24,24,24,24,0
202 data■■6,■■6,■■6,98,98,98,126,0
204 data134,70,70,38,■■2,■■2,14,0
206 data98,98,98,98,■■6,90,102,0
208 data■■6,36,24,124,98,98,98,0
210 data■■6,■■6,■■6,60,24,24,24,0
212 data2■■5,■■27,6,28,48,■■12,2■■5,0
248 data2■■5,0,0,0,0,0,0,0
250 data0,0,0,0,0,0,0,2■■5
252 data128,128,128,128,128,128,128,128
254 data1,1,1,1,1,1,1,1
256 data2■■5,128,128,128,128,128,128,128
258 data0,0,0,0,0,0,0,0
260 data2■■5,1,1,1,1,1,1,1
262 data0,38,38,38,0,0,0,0
264 data128,128,128,128,128,128,128,2■■5
266 data1,1,1,1,1,1,1,2■■5
268 data192,48,12,3,0,0,0,0
270 data0,0,0,0,192,48,12,3
272 data0,0,0,0,3,12,48,192
274 data3,12,48,192,0,0,0,0
276 data128,128,64,64,32,32,16,16
278 data8,8,4,4,2,2,1,1
280 data1,1,2,2,4,4,8,8
282 data16,16,32,32,64,64,128,128
284 data2■■5,176,140,131,128,128,128,128
285 data2■■5,13,49,193,1,1,1,1
286 data1■■4,1■■4,160,160,192,192,128,2■■5
289 rem■SPACE■data■SPACE■for■SPACE■numbers
290 data254,130,134,210,■■26,194,254,0
291 data■■12,16,16,48,48,48,120,0
292 data254,130,2,254,192,192,254,0
293 data126,2,2,62,6,6,254,0
294 data128,128,132,132,254,12,12,0
295 data126,64,64,126,6,70,254,0
296 data254,128,128,254,194,194,254,0
297 data254,130,4,8,24,24,24,0
298 data120,68,68,254,194,194,254,0
299 data254,130,130,254,6,6,254,0
303 data9,9,5,5,3,3,1,2■■5
304 data128,128,128,128,131,140,176,192
305 data1,1,1,1,193,49,13,3
306 data2■■5,0,0,0,192,48,12,3
307 data2■■5,0,0,0,3,12,48,192
308 data129,129,130,130,132,132,136,136
309 data129,129,65,65,■■3,■■3,17,17
310 data0,0,0,0,0,3,15,31
315 data0,0,3,63,2■■5,2■■5,2■■5,2■■5
320 data0,31,2■■5,2■■5,2■■5,2■■5,2■■5,2■■5
325 data63,2■■5,2■■5,2■■5,2■■5,2■■5,2■■5,2■■5
330 data252,2■■5,2■■5,2■■5,2■■5,2■■5,2■■5,2■■5
335 data0,248,2■■5,2■■5,2■■5,2■■5,2■■5,2■■5
340 data0,0,192,252,2■■5,2■■5,2■■5,2■■5
345 data0,0,0,0,0,192,240,248
350 data63,127,2■■5,2■■5,2■■5,2■■5,127,63
355 data252,254,2■■5,2■■5,2■■5,2■■5,252,252
360 data31,15,3,0,0,0,0,0
365 data2■■5,2■■5,2■■5,2■■5,63,3,0,0
370 data2■■5,2■■5,2■■5,2■■5,2■■5,2■■5,31,0
375 data2■■5,2■■5,2■■5,2■■5,2■■5,2■■5,2■■5,63
380 data2■■5,2■■5,2■■5,2■■5,2■■5,2■■5,2■■5,252
385 data2■■5,2■■5,2■■5,2■■5,2■■5,2■■5,248,0
390 data2■■5,2■■5,2■■5,2■■5,252,192,0,0
395 data248,240,192,0,0,0,0,0
400 data2■■5,2■■5,2■■5,2■■5,2■■5,2■■5,2■■5,2■■5
415 data0,0,0,0,0,3,7,15
416 data0,0,3,63,2■■5,2■■5,2■■5,2■■5
417 data0,63,2■■5,2■■5,2■■5,2■■5,2■■5,2■■5
420 data0,252,2■■5,2■■5,2■■5,2■■5,2■■5,2■■5
425 data0,0,192,252,2■■5,2■■5,2■■5,2■■5
430 data0,0,0,0,0,192,■■24,240
435 data15,7,3,0,0,0,0,0
440 data2■■5,2■■5,2■■5,2■■5,63,3,0,0
445 data2■■5,2■■5,2■■5,2■■5,2■■5,2■■5,63,0
450 data2■■5,2■■5,2■■5,2■■5,2■■5,2■■5,252,0
455 data2■■5,2■■5,2■■5,2■■5,252,192,0,0
460 data240,■■24,192,0,0,0,0,0
480 rem■SPACE■data■SPACE■for■SPACE■disc
490 data0,2■■5,2■■5,2■■5,2■■5,2■■5,2■■5,2■■5
495 data0,240,252,254,254,2■■5,2■■5,2■■5
500 data195,129,129,0,0,0,0,0
505 data0,0,0,0,1,1,3,7
510 data2■■5,2■■5,2■■5,254,254,252,248,■■24
515 data15,15,15,15,15,15,15,15
520 data2■■5,254,254,252,252,240,■■24,0
525 data0,0,0,■■12,248,248,252,252
530 data254,254,2■■5,2■■5,2■■5,2■■5,2■■5,2■■5
535 data3,15,63,127,127,2■■5,2■■5,2■■5
540 data240,248,252,252,252,252,248,240
545 data2■■5,2■■5,2■■5,2■■5,2■■5,2■■5,2■■5,126
550 data128,0,0,0,0,0,63,63,63
555 data0,0,0,0,0,240,248,248
560 data63,63,63,0,0,0,0,0
565 data252,252,254,254,126,126,126,254
568 data254,254,252,252,248,248,240,■■24
570 data0,7,15,31,31,63,63,63
575 data0,2■■5,2■■5,2■■5,2■■5,2■■5,2■■5,2■■5
580 data0,252,252,252,252,252,252,252
585 data63,63,63,63,63,63,63,63
590 data■■24,■■24,■■24,■■24,■■24,■■24,■■24,■■24
595 data■■24,■■24,■■24,■■24,■■24,■■24,240,248
600 data63,31,31,31,15,15,7,1
605 data252,252,252,252,252,252,252,252
610 data127,2■■5,2■■5,2■■5,2■■5,2■■5,127
615 data1,3,7,15,31,63,127,2■■5
620 data128,192,■■24,240,248,252,254,2■■5
625 data31,31,31,31,31,0,0,0
630 data248,248,248,248,248,0,0,0
800 rem■SPACE■data■SPACE■for■SPACE■tron
810 v=53248
820 poke50168,18
830 forn=0to62:readq:poke50304+n,q:next
840 data0,24,0,0,60,0,0,126,0,3,2■■5,192,7,195,■■
24,7,219,■■24,14,219,■■12
850 data14,231,■■12,15,126,240,7,126,■■24,0,126,0
,0,126,0,0,126,0,0,2■■5,0
860 data0,231,0,1,231,128,1,231,128,1,195,128,3,1
95,192,3,129,192,3,129,192
870 pokev+39,7:pokev+0,140:pokev+1,1■■4
900 rem■SPACE■data■SPACE■for■SPACE■zark
910 poke50169,19
920 forn=0to62:readq:poke50368+n,q:next
930 data0,0,0,0,0,0,0,0,0,0,24,0,0,60,0,0,102,0,1
,2■■5,128,3,2■■5,192
940 data3,2■■5,192,7,126,■■24,7,126,■■24,3,189,19
2,1,189,128,0,126,0,0,126,0
950 data0,2■■5,0,0,231,0,1,195,128,1,195,128,3,12
9,192,3,129,192
960 pokev+2,140:pokev+3,104
970 pokev+40,15
1000 rem■SPACE■data■SPACE■for■SPACE■disc
1005 poke50170,20
1010 forn=0to62:readq:poke50432+n,q:next
1015 data0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,60,0,0,2■■5,0,1,2■■5,128
1020 data0,2■■5,0,0,60,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1025 pokev+41,1:pokev+4,154:pokev+5,103
1030 poke50171,20:pokev+42,1:pokev+6,151:pokev+7,
140:rem■SPACE■disk■SPACE■2
1100 rem■SPACE■data■SPACE■for■SPACE■c■■ordinater
1110 poke50172,21
1120 forn=0to62:readq:poke50496+n,q:next
1130 data0,0,0,0,0,0,1,2■■5,192,0,2■■5,128,0,2■■5
,128,0,127,0,0,127,0,0,62,0
1140 data0,28,0,0,28,0,0,8,0,0,28,0,0,28,0,0,62,0
,0,127,0,0,127,0,0,2■■5,128
1150 data0,2■■5,128,1,2■■5,192,0,0,0,0,0,0
1160 pokev+43,5:pokev+8,107:pokev+9,95
1800 rem■SPACE■data■SPACE■for■SPACE■variabLes
1805 x1=107:y1=95:x=152:y=140:n=6:zx=154:zy=103:z
n=6:ab=0:v=53248
1810 fori=1to5:hs(i)=0:next
2000 print"■shCLR■"
2010 print"■■■SPACE■am■■c■SPACE■power■SPACE■c■■o
Lware■SPACE■@"
2020 print"■■■SPACE■presents■DOWN■"
2030 print"■■■SPACE■discs■SPACE■of■SPACE■tron"
2040 print"■■■SPACE■■■■DOWN■he■■Lo■SPACE■user"
2050 print"■■■SPACE■you■SPACE■are■SPACE■in■■■SPAC
E■on■SPACE■Line"
2060 print"■■■SPACE■set■SPACE■LeveL■SPACE■on■SPAC
E■next■SPACE■page"
2070 print"■DOWN■■■■SPACE■hit■SPACE■any■SPACE■ke
y"
2072 print"■■■DOWN■■■■SPACE■Joystick":print"■■■SP
```

# C64 Listings

```
ACE■for■SPACE■port■SPACE■2"
 2080 geta$:ifa$=""then2080
 2200 print"■shCLR■";
 2205 printtab(■1)"■c=7■■■shSPACE■■c=k■■SPACE■■c
=m■■SPACE■■shPOUND■■shs■■c=n■■c=e■■c=r■■c=w■■SPAC
E■■shPOUND■■shs■■c=n■"
 2210 printtab(■1)"■c=7■■shs■■c=o■■shs■■SPACE■■c
=POUND■■SPACE■■c=q■■c=d■■c=z■■c=h■■c=j■■■SPACE■■c
=q■■c=d■■c=z■"
 2215 printtab(■1)"■ct5■■shs■■c=t■■shs■■SPACE■■sh
s■■■SPACE■■c=s■■c=@■■c=h■■c=L■■■SPACE■■c=s■■c=@■
"
 2220 printtab(■1)"■c=3■■■c=g■■c=+■■SPACE■■shs■■
SPACE■■c=p■■shs■■c=a■■c=y■■shs■■c=u■■SPACE■■c=p■■s
hs■■c=a■"
 2225 printtab(18)"■DOWN■of"
 2230 printtab(17)"■DOWN■tron"
 2300 printtab(14)"■■■DOWN■created■SPACE■by"
 2310 printtab(8)"the■SPACE■am■■c■SPACE■power■■■SP
ACE■boc"
 2315 printtab(8)"c■■oLware■SPACE■team■■■SPACE■mer
t"
 2330 print"■■■DOWN■■■■SPACE■game■■■■SPACE■instru
ction"
 2335 print"■■■SPACE■pLay■■■■SPACE■set"
 2340 printtab(5)"■c=7■■sh@■■c=f■"
 2345 printtab(5)"■c=c■■c=x■"
 2348 rem■SPACE■abfrage■SPACE■fuer■SPACE■joystick
 2350 j=peek(56320)
 2355 if(jand8)=0thenin=1
 2358 if(jand4)=0thenin=0
 2360 print"■HOME■■■■■DOWN■";
 2370 ifin=1then2380
 2372 printtab(5)"■sh@■■c=f■":printtab(5)"■c=c■■c=
x■"
 2374 fori=1to20:if(jand16)=0then4■■0
 2376 next:print"■HOME■■■■■DOWN■"tab(5)"■■■SPACE■
":printtab(5)"■■■SPACE■":gosub402■■0
 2378 goto2350
 2380 printtab(29)"■sh@■■c=f■":printtab(29)"■c=c■■
c=x■"
 2384 fori=1to20:if(jand16)=0thengosub605■■0:in=0:
j=0:goto■■2■■0
 2386 next:print"■HOME■■■■■DOWN■"tab(29)"■■■SPACE■
":printtab(29)"■■■SPACE■":gosub402■■0
 2388 goto2350
 4000 print"■shCLR■";
 4010 print"■■■SPACE■-=■■■{>."
 4020 print"■■■SPACE■)■SPACE■%&■■■SPACE■'(■SPACE■
~"
 4030 print"■■■SPACE■)■■■SPACE■%&■■■■SPACE■'(■■■S
PACE■~"
 4040 print"■■■SPACE■)■■■SPACE■■ARROWLEFT■■■■{!■■■
SPACE■~"
 4050 print"■■■SPACE■)■■■SPACE■)■■■■SPACE■~■■■SPA
CE■~"
 4060 print"■■■SPACE■)■■■SPACE■)■■■■SPACE■~■■■SPA
CE■~"
 4070 print"■■■SPACE■)■■■SPACE■#■■■■POUND■$■■■SPAC
E■~"
 4080 print"■■■SPACE■)■■■SPACE■'()■■■■SPACE■~%&■■■
SPACE■~"
 4090 print"■■■SPACE■)■SPACE■'(■■■SPACE■)■■■■SPAC
E■~■■■SPACE■%&■SPACE■~"
 4100 print"■■■SPACE■;(■■■SPACE■)■■■■SPACE■~■■■SP
ACE■%<"
 4110 print"■■■SPACE■)■■■SPACE■#■■■■POUND■$■■■SPA
CE■~"
 4120 print"■■■SPACE■)■■■SPACE■+■■■■SPACE■)■■■SPA
CE■~"
 4130 print"■■■SPACE■)■■■SPACE■,■■■■SPACE■*■■■SPAC
E■~"
 4140 print"■■■SPACE■)■■■SPACE■+■■■■SPACE■)■■■SPA
CE■~"
 4150 print"■■■SPACE■)■■■SPACE■,■■■■SPACE■*■■■SPA
CE■~"
 4160 print"■■■SPACE■)■■■SPACE■+■■■■SPACE■)■■■SPA
CE■~"
 4170 print"■■■SPACE■)■■■SPACE■,■■■■SPACE■*■■■SPAC
E■~"
 4180 print"■■■SPACE■)■SPACE■+■■■■SPACE■)■SPACE■~
"
 4190 print"■■■SPACE■)■SPACE■,■■■■SPACE■*■SPACE■~
 4200 print"■■■SPACE■)+■■■■SPACE■)~"
 4210 print"■■■SPACE■),■■■■SPACE■*~"
 4220 print"■■■SPACE■?■■■■SPACE■■sh*■"
 4230 print"■■■SPACE■/■■■■POUND■:"
 5000 rem
 6000 print"■HOME■■ct5■"tab(29)"tron■SPACE■2■■■SP
ACE■by"
 6010 printtab(29)"■SPACE■c■■oLware@"
 6020 printtab(29)"■DOWN■on■SPACE■Line"
 6040 printtab(29)"■DOWN■zark"
 6060 printtab(29)"tron"
 6070 printtab(29)"■DOWN■LeveL■SPACE■01"
 6080 printtab(29)"disc■SPACE■02"
 6090 printtab(29)"system■SPACE■rts"
 6100 printtab(29)"■SPACE■active■SPACE■02"
 6110 printtab(29)"■SPACE■sys■SPACE■0640"
 6120 printtab(29)"■DOWN■men"
 6500 gosub■■5■■0
 8000 rem■SPACE■joystick■SPACE■abfrage■SPACE■fuer■
SPACE■spieL
 8002 j=peek(56320)
 8005 ifh4=1thengoto18■■0
 8006 ifh3=1orrnd(1)>.5thengosub5■■0
 8007 ifh3=1orrnd(1)>.8thengoto17■■0
 8008 ify+3>zyandy-3<zyandx+20>zxandx<zx+20thengot
o2■■0
 8009 ify+3>zyandy-3<zyandx-20>zxandx<zx-20thengot
o2■■0
 8010 ifh3=1thengoto15■■0
 8011 if(jand16)=0thengoto14■■0
 8014 ifj=121orj=■■17then54■■0
 8015 if(jand8)=0thenx1=x1+8
 8020 if(jand4)=0thenx1=x1-8
 8030 ifx1<28thenx1=x1+8:goto8■■0
 8040 ifx1>250thenx1=x1-8:goto8■■0
 8100 ifx1<83orh1=1thengoto■■1■■0
 8110 ifx1>195orh2=1thengoto1■■0
 9000 pokev+8,x1:pokev+9,y1
 9070 ify1=95thenh1=0:h2=0
 9100 goto8■■0
 10000 rem■SPACE■moving■SPACE■diagonaL■SPACE■righ
t
 10050 if(jand8)=0theny1=y1+4
 10060 if(jand4)=0theny1=y1-4
 10100 h2=1:goto9■■0
 11000 rem■SPACE■moving■SPACE■diagonaL■SPACE■Left
 11010 if(jand8)=0theny1=y1-4
 11020 if(jand4)=0theny1=y1+4
 11500 h1=1:goto9■■0
 14000 rem■SPACE■tron■SPACE■schu■■s
 14010 n=7-abs■■(95-y1)/4.8)
 14020 dm=d6+t1:x=d6+t1:y=140
 15000 rem■SPACE■schu■■s
 15005 y2=y1+6
 15010 ify<=y2thenn=-n
 15020 x=x+■■(x1-dm)/(140-y2■■)*n
 15030 y=y-n
 15080 ify>140thenh3=0:n=7:goto8■■0
 15110 ify-4<v3andy>v3andx-12<v2+z1andx-12>v2-20+z
1thengoto25■■0
 15500 pokev+6,x:pokev+7,y
 16000 h3=1:goto8■■0
 17000 rem■SPACE■zark■SPACE■schu■■s
 17020 ifh3=1thenz1=v0+t1:goto17060
 17030 ifrnd(1)<.7andh3=0thenz1=v0+t1:goto17060
 17040 ifrnd(1)>.4thengoto8■■0
 17050 z1=rnd(1)*70+1■■0
 17060 z2=150
 17070 zp=d4+z1:zx=d4+z1:zy=103
 17100 nz=8-abs■■(150-z2)/7.4)
 18000 rem■SPACE■schu■■s■SPACE■zark
 18050 ifzy>z2thennz=-nz
 18080 zx=zx+■■(z1-zp)/(z2-103■■)*nz
 18090 zy=zy+nz
 18110 ifzy<103thenh4=0:goto8■■08
 18125 ifzy+4>v1andzy<v1andzx-12<v0+t1andzx-12>v0-
20+t1thengoto26■■0
 18150 pokev+4,zx:pokev+5,zy
 18200 h4=1:goto8■■08
 20000 rem■SPACE■disc■SPACE■crash
 20010 pokev+21,19
 20020 forx5=15to0step-1:poke54296,x5:poke542■■7,9
```

```
:poke54278,15:poke54273,40
 20030 poke54272,2■■5:poke54276,129:next:poke54276
,0:poke542■■7,0
 20100 x=152:y=140:pokev+6,x:pokev+7,140:zx=154:zy
=103:pokev+4,zx:pokev+5,zy
 20110 pokev+21,31
 23000 h4=0:h3=0:ct=ct+50:goto8■■0
 24999 rem[SPACE]zark[SPACE]Lo■■ged[SPACE]o■■f
 25000 da=15:fori1=1to15:pokev+40,i1:pokev+3,v3+i1
*2:da=da-1
 25010 poke54296,15:poke542■■7,9:poke54278,2:poke5
4273,40+da*1.3
 25020 poke54272,1■■0:poke54276,17:poke54276,0:po
ke54276,17:next
 25040 poke54276,0:poke542■■7,0
 25050 fori=1to4■■0:next
 25080 pokev+21,19
 25110 gosub6■■0:pokev+21,31
 25200 h3=0:h4=0:ct=ct+3■■0:ab=ab+1:gosub3■■0
 25205 ifm=4thengoto■■2■■0
 25210 goto8■■0
 25999 rem[SPACE]tron[SPACE]Lo■■ged[SPACE]o■■f
 26000 da=15:fori1=1to15:pokev+39,i1:pokev+1,v1+i1
*2:da=da-1
 26010 poke54296,15:poke542■■7,9:poke54278,2:poke5
4273,40+da*1.3
 26020 poke54272,1■■0:poke54276,17:poke54276,0:pok
e54276,17:next
 26040 poke54276,0:poke542■■7,0
 26050 fori=1to3■■0:next
 26080 pokev+21,19
 26110 gosub601■■0
 26200 pokev+21,31:h3=0:h4=0:cz=cz+3■■0:m=m+1:gosu
b3■■0
 26280 ifm=4thengoto■■2■■0
 26300 goto8■■0
 27030 poke54296,15:poke542■■7,29:poke54278,16:pok
e54273,20
 30000 rem[SPACE]counting
 30010 ford=0toctstep50
 30020 print"[HOME]■■[DOWN]■[ct2]";tab(■■3);rt+d:g
osub4■■0:next
 30030 fori=1to1■■0:next
 30040 ford=0toczstep50
 30050 print"[HOME]■■[DOWN]";tab(■■3);rz+d:gosub4■
■0:next
 30100 rt=rt+ct:rz=rz+cz:ct=0:cz=0
 30200 print"[ct5]■■[DOWN]"tab(■■3);4-m
 30220 if4-m=0then32■■0
 30230 ifab=5thengoto565■■0
 31000 return
 32000 rem■■[SPACE]game[SPACE]over
 32005 fori=1to3■■0:next
 32010 print"[HOME]■■[DOWN]■[ct2]"tab(12)"game[SPA
CE]over"
 32020 gosub4■■0
 32100 fori=1to1■■0:next:pokev+21,0:ab=0:gosub62■■
0:return
 40000 rem[SPACE]sound[SPACE]by[SPACE]writing
 40020 poke54296,15:poke542■■7,0:poke54278,16:poke
54273,10
 40030 poke54272,69:poke54276,129
 40040 poke54276,0:poke542■■7,0
 40100 return
 40200 rem[SPACE]p■■ep[SPACE]sound
 40220 poke54296,15:poke542■■7,0:poke54278,96:poke
54273,60
 40230 poke54272,■■3:poke54276,17:fori=1to120:next
 40240 poke54276,0:poke542■■7,0
 40300 return
 40400 rem[SPACE]sound[SPACE]for[SPACE]disc[SPACE]
a■■pearing
 40410 poke54296,15:poke542■■7,29:poke54278,16:pok
e54273,20
 40420 poke54272,1■■0:poke54276,129:fors1=1to50:ne
xt
 40430 poke54276,0:poke542■■7,0
 40440 return
 50000 rem[SPACE]zark[SPACE]moving
 50020 ifrnd(1)>.6andzl<bzthenzl=zl+7:goto5■■050
 50030 ifzl>-gzthenzl=zl-7
 50050 pokev+2,v2+zl:pokev+4,d4+zl
 51000 return
 54000 rem[SPACE]tron[SPACE]moving
 54010 ifj=■■17andtl<btthentl=tl+8
 54020 ifj=121andtl>-gtthentl=tl-8
 54050 pokev+0,v0+tl:pokev+6,d6+tl
 54200 goto8■■0
 55000 rem[SPACE]spieLaufbau
 55010 rem
 55100 ifab=5then56510
 56000 fori=1to6■■0:next
 56010 print"[HOME]■■[DOWN]■■[RIGHT]■[ct5]";
 56020 print"[sha]■[shb]■[shc]■[shd]■[she]■[shf]■[shg]■[s
hh]■"
 56030 print"■■[RIGHT]■[shi]■■[shs]■[shj]■"
 56040 print"■■[RIGHT]■[shk]■[shL]■[shm]■[shn]■[sho]■
shp]■[shq]■[shr]■":gosub404■■0:fori=1to3■■0:next
 56050 print"[HOME]■■[DOWN]■■[RIGHT]■";
 56060 print"[sht]■[shu]■[shv]■[shw]■[shx]■[shy]■"
 56070 print"■■[RIGHT]■[shz]■[sh+]■[c=-]■[sh-]■[c=■]■[c
=*]■":gosub404■■0
 56100 v0=140:v1=1■■4:v2=140:v3=104:d4=154:d5=103:
d6=153:d7=140:da=0
 56105 bt=24:bz=13:gt=24:gz=13:rt=0:rz=0:m=0:ab=0
 56110 pokev+21,1:gosub601■■0
 56120 fori=1to3■■0:next
 56130 pokev+21,3:gosub6■■0
 56150 pokev+21,31:gosub4■■0
 56200 fori=1to2■■0:next:gosub59■■0:fori=1to5■■0:n
ext
 56400 return
 56500 rem[SPACE]runde[SPACE]2
 56505 rem[SPACE]gosub62■■0[SPACE]sound
 56506 pokev+21,0:goto4■■0
 56510 fori=1to6■■0:next
 56520 print"[HOME]■■[DOWN]■";
 56530 printtab(7)"[sha]■[shb]■[shc]■[shd]■[she]■[shf]■
shg]■[shh]■"
 56540 printtab(7)"[shi]■■[shs]■[shj]■"
 56550 printtab(7)"[shk]■[shL]■[shm]■[shn]■[sho]■[shp]■
shq]■[shr]■":gosub404■■0
 56560 print"[HOME]■■[DOWN]■";
 56570 printtab(17)"[sha]■[shb]■[shc]■[shd]■[she]■[shf]■
[shg]■[shh]■"
 56580 printtab(17)"[shi]■■[shs]■[shj]■"
 56590 printtab(17)"[shk]■[shL]■[shm]■[shn]■[sho]■[shp]■
[shq]■[shr]■":gosub404■■0
 56600 print"[HOME]■■[DOWN]■";
 56610 printtab(16)"[sht]■[shu]■[shv]■[shw]■[shx]■[shy]■
"
 56620 printtab(16)"[shz]■[sh+]■[c=-]■[sh-]■[c=■]■[c=*]■
":gosub404■■0
 56630 print"[HOME]■■[DOWN]■";
 56640 printtab(10)"[sht]■[shu]■[shv]■[shw]■[shx]■[shy]■
"
 56650 printtab(10)"[shz]■[sh+]■[c=-]■[sn-]■[c=■]■[c=*]■
":gosub404■■0
 56680 v0=180:v1=1■■4:v2=■■16:v3=104:d4=135:d5=103
:d6=194:d7=140:da=0
 56685 bt=25:bz=50:gt=■■12:gz=13:ab=0:m=0
 56690 pokev+21,1:gosub601■■0:pokev+21,3:gosub6■■0
 56700 pokev+21,31:gosub59■■0:fori=1to5■■0:next
 56710 return
 56900 return
 59000 rem[SPACE]start
 59010 print"[HOME]■■[DOWN]■";
 59030 printtab(31)"pre■■s"
 59040 printtab(31)"fire[SPACE]"
 59052 fori=1to5:if(peek(56320)and16)=0thenreturn
 59065 next:gosub402■■0:print"[HOME]■■[DOWN]■";
 59090 printtab(31)"■■[SPACE]"
 59100 printtab(31)"■■[SPACE]"
 59110 fori=1to10
 59115 if(peek(56320)and16)=0thenreturn
 59120 next
 59150 goto59010
 60000 rem[SPACE]zark[SPACE]transport
 60010 pokev+2,v2:pokev+3,v3
 60020 da=0:fori5=1to15:pokev+40,i5:da=da+1
 60030 poke54296,15:poke542■■7,9:poke54278,2:poke5
4273,40+da*1.3
 60040 poke54272,1■■0:poke54276,17:poke54276,0:pok
e54276,17:next
 60050 poke54276,0:poke542■■7,0
 60070 pokev+2,v2:pokev+3,v3:pokev+4,d4:pokev+5,d5
```

# C64 Listings

```
:pokev+6,d6+t1:pokev+7,d7
 60075 x=d6+t1:y=140:zx=d4:zy=d5
 60080 pokev+40,15:z1=0
 60090 return
 60100 rem SPACE tron SPACE transport
 60110 pokev+0,v0:pokev+1,v1:pokev+4,d4:pokev+5,d5
:pokev+6,d6:pokev+7,d7:da=0
 60120 fori=1to15:pokev+39,i6:da=da+1
 60130 poke54296,15:poke542??7,9:poke54278,2:poke5
4273,40+da*1.3
 60140 poke54272,1??0:poke54276,17:poke54276,0:pok
e54276,17:next
 60150 poke54276,0:poke542??7,0
 60175 x=d6:y=140:zx=d4+z1:zy=d5
 60180 pokev+39,7:t1=0
 60200 return
 60500 rem SPACE instruction SPACE set
 60510 print"shCLR ct2 ??? SPACE instructions"
 60512 print"DOWN c=7 ??? SPACE sie SPACE sind S
PACE der":print"?? SPACE spieLer SPACE tron":print
"??? SPACE in SPACE ct8 geLb"
 60514 printtab(??2)"??? DOWN c=7 und SPACE spieLe
n":printtab(??2)"gegen SPACE ihren"
 60516 printtab(??2)"gegner SPACE zark":printtab(?
?2)"in SPACE c=8 grau"
 60530 print"??? DOWN c=7 ??? SPACE ihr SPACE spieL
er SPACE befindet SPACE sich SPACE auf"
 60532 print"??? SPACE der SPACE vorderen SPACE sc
heibe SPACE und SPACE ihr SPACE gegner"
 60534 print"??? SPACE ist SPACE auf SPACE der SPA
CE hinteren SPACE scheibe"
 60536 print"??? SPACE zieL SPACE des SPACE spieLes
 SPACE ist SPACE es SPACE mit SPACE ihrer"
 60538 print"??? SPACE ct2 disc c=7 ? SPACE zark SP
ACE zu SPACE tre??fen SPACE und SPACE ihn SPACE so
"
 60540 print"??? SPACE von SPACE der SPACE scheibe
 SPACE zu SPACE schmei??sen"
 60542 print"??? DOWN ??? SPACE pre??s SPACE fire SP
ACE for SPACE next SPACE page"
 60543 poke53248,175:poke53249,70:poke53250,150:po
ke53251,120:poke53248+21,3
 60544 fori=1to5??0:next
 60545 j=peek(56320)
 60548 if(jand16)=0thenpoke53248+21,0:goto60??5
 60550 goto60545
 60555 print"shCLR ct2 ??? SPACE instruction c=7 ?
??? SPACE page SPACE 2"
 60558 print"DOWN um SPACE auf SPACE zark SPACE z
ieLen SPACE zu SPACE koe??nen SPACE steht"
 60560 print"ihnen SPACE ein SPACE zieLkreuz SPACE
 zur SPACE verfuegung"
 60562 print"weLches SPACE sie SPACE nach SPACE Li
nks SPACE oder SPACE nach"
 60564 print"rechts SPACE mit SPACE hiLfe SPACE des
 SPACE Joysticks"
 60570 print"bewegen SPACE koe??nen":print"druecke
n SPACE sie SPACE nun SPACE den SPACE feuerknopf SP
ACE so"
 60572 print"bewegt SPACE sich SPACE die SPACE disc
 SPACE genau SPACE auf SPACE das"
 60574 print"zieLkreuz SPACE zu SPACE und SPACE pra
??Lt SPACE von SPACE der SPACE wand"
 60576 print"wieder SPACE ab SPACE und SPACE beweg
t SPACE sich SPACE zum SPACE spieLer"
 60578 print"zurueck"
 60580 print"DOWN pa??sen SPACE sie SPACE jedoch
SPACE auch SPACE auf SPACE zark"
 60582 print"da SPACE dieser SPACE auch SPACE in
SPACE der SPACE Lage SPACE ist SPACE auf":print"sie
 SPACE zu SPACE schie??sen"
 60584 print"DOWN gLeuckLicherweise SPACE sind SP
ACE sie SPACE jedoch"
 60590 print"in SPACE der SPACE Lage SPACE ihm SPA
CE auszuweichen"
 60592 print"indem SPACE sie SPACE ihren SPACE Joy
stick SPACE benutzen"
 60594 print"DOWN ??? SPACE pre??s SPACE fire SPA
CE for SPACE next SPACE page"
 60600 j=peek(56320)
 60602 if(jand16)=0then60630
 60604 goto606??0
 60630 print"shCLR ct2 ??? SPACE instructions c=7
??? SPACE page SPACE 3"
 60635 print"??? DOWN sie SPACE mue??sen SPACE einf
ach"
 60638 print"den SPACE Joystick SPACE nach SPACE L
inks SPACE unten SPACE bewegen"
 60640 print"um SPACE tron SPACE nach SPACE Links
SPACE zu SPACE bewegen"
 60642 print"??? SPACE oder":print"den SPACE Joysti
ck SPACE nach SPACE rechts SPACE unten SPACE beweg
en"
 60644 print"um SPACE nach SPACE rechts SPACE zu S
PACE geLangen"
 60648 print"DOWN sie SPACE koe??nen SPACE tron S
PACE jedoch SPACE nur SPACE bewegen"
 60650 print"we??n SPACE sie SPACE nicht SPACE ger
ade SPACE schie??sen"
 60655 print"ebenso SPACE koe??nen SPACE sie SPAC
E sich SPACE nur SPACE auf SPACE der"
 60660 print"scheibe SPACE bewegen SPACE und SPACE
 nicht SPACE au??serhaLb":print"davon"
 60664 print"haben SPACE sie SPACE jedoch SPACE za
rk SPACE fuenf SPACE maL"
 60668 print"abgescho??sen SPACE so SPACE ko??men
SPACE sie SPACE in SPACE die SPACE"
 60670 print"naechste SPACE runde SPACE wo SPACE
vier SPACE scheiben"
 60672 print"erscheinen SPACE auf SPACE denen SPAC
E sie SPACE sich SPACE bewegen":print"koe??nen"
 60674 print"DOWN ??? SPACE pre??s SPACE fire SPAC
E for SPACE next SPACE page"
 60680 j=peek(56320)
 60684 if(jand16)=0thengoto607??0
 60690 goto60680
 60700 print"shCLR ct2 ??? SPACE instructions c=7
??? SPACE page SPACE 4"
 60705 print"??? DOWN so??Lten SPACE sie SPACE das
SPACE progra??m SPACE waehrend"
 60710 print"des SPACE Laufens SPACE sto??pen SPA
CE sol SPACE ist SPACE es SPACE moegLich"
 60715 print"das SPACE das SPACE progra??m SPACE f
ehLerhaft SPACE Laeuft"
 60720 print"we??n SPACE sie SPACE es SPACE wieder
SPACE mit SPACE dem SPACE befehL"
 60725 print"ct2 continue c=7 ? SPACE weiterLaufen
SPACE La??sen"
 60730 print"deswegen SPACE ist SPACE es SPACE rat
sam SPACE nach SPACE dem"
 60735 print"sto??pen SPACE des SPACE progra??ms S
PACE die SPACE tasten"
 60740 print"??? SPACE ct2 runstop SPACE restore c
=7 ?"
 60745 print"zu SPACE druecken SPACE und SPACE dan
ach SPACE den SPACE befehL"
 60750 print"??? SPACE ct2 poke SPACE 648,4 c=7 ?"
 60755 print"einzugeben SPACE danach SPACE befinde
t SPACE sich SPACE der"
 60760 print"rechner SPACE wieder SPACE im SPACE a
rbeitszustand SPACE und"
 60770 print"das SPACE progra??m SPACE ka??nn SPACE
 durch SPACE ct2 run c=7 ? SPACE wieder"
 60775 print"gestartet SPACE werden"
 60800 print"DOWN ??? SPACE pre??s SPACE fire SPAC
E to SPACE return SPACE to SPACE game"
 61010 j=peek(56320)
 61020 if(jand16)=0thenreturn
 61030 goto61010
 62000 rem SPACE high SPACE score
 62010 print"shCLR";
 62020 printtab(12)"high SPACE score"
 62040 print"??? DOWN ":fort=1to5
 62050 printtab(10)t;" SPACE score"tab(20)hs(t)
 62060 next
 62080 ifrt>hsthen623??0
 62100 goto625??0
 62300 fori=5to1step-1:ifrt>hs(i)thenhb=i
 62310 next
 62350 forn=1to1??0:next:hs(hb)=rt
 62380 rt=0:goto62010
 62500 print"HOME ??? DOWN ??? SPACE pre??s SPACE
fire SPACE to SPACE return SPACE to SPACE game"
 62510 j=peek(56320)
 62520 if(jand16)=0thenreturn
 62550 goto62510??
```

AMIGA Public Domain (FAUG usw)
10 Disks 45 DM 20 Disks 85 DM
C 64 = DFB-Pokal (Game auf Disk)
nur 10 DM P. Keim, Vogelsanger
Str. 34, 5 Köln 30, T=0221/520765

---

| | |
|---|---|
| JUPITERSOFT PRES. SOFTWARE | C 64 |
| HEXEN MÜSSEN BRENNEN | Disk 15.90 |
| DAS WAR DER WILDE WESTEN | Disk 15,90 |
| oder kolo. | Info anfordern |
| Höttinger Str. 34 b | Comp.- Typ |
| 8836 Ellingen!!! | angeben!! |

---

**AMIGA HARDWARE - SOFTWARE AMIGA**
**C 64/128** Zubehör **C 64/128**

| | | |
|---|---|---|
| AMIGA 2000 3,5" Laufwerk | nur | 269,00 |
| Amiga 3,5" Zweitlaufwerk | nur | 349,00 |
| SWITCH JOY Mikroschal. | nur | 19,00 |
| Disketten 3,5" MF2DD | nur | 27,00 |
| No Name MD1D 10 Stück | nur | 7,50 |

Drucker, Monitore, Computer,
Kabel usw. günstig lieferbar!
Versand per NN plus Versandkosten
Bitte kostenlose Info anfordern!
**Maske - Computer - Elektronik**
Ernst Walter Maske
Neuer Weg 15, 2060 Sülfeld

Computer-Flohmarkt
Interessenten erhalten die Gelegenheit, Computer, Peripherie, Bausteine und Literatur, alles was zum Computer gehört, gebraucht zu kaufen bzw. zu verkaufen. Die Standgebühr für einen Tisch von ca. 0,60 x 1,20 m beträgt für Privatleute DM 10,00 (Schüler DM 5,00) und für Händler DM 20,00. Bei Vorlage eines Parkscheins wird eine Ermäßigung von DM 4,00 gewährt.
Um eine einwandfreie Organisation zu gewährleisten, werden die Verkäufer gebeten, sich bis zum 1. Februar 1988 schriftlich bei Burkhard John, Harleßstraße 9, 4000 Düsseldorf 1, Tel.: 0211 / 63 29 78 anzumelden.
Samstag, 6. Februar 9.00 - 15.00 Uhr
WEITERBILDUNGSZENTRUM am Hauptbahnhof, Bertha-von-Suttner-Platz 1, Foyer, gebührenfrei.

Veranstaltungen auf dem Computer-Flohmarkt
(Vortrag) Programmiersprachen im Vergleich.
Dr. Klaus Eyerszumrode
Übersicht über gebräuchliche Programmiersprachen und Vergleich im Hinblick auf die Einsatzmöglichkeiten.
Samstag, 6. Februar, 10.00 Uhr, Saal 2
WEITERBILDUNGSZENTRUM am Hauptbahnhof, Bertha-von-Suttner-Platz 1, Saal 2, gebührenfrei

(Vortrag) Zählt Ihr Computer morgen schon zum alten Eisen?
Eduard Altmann
Brandheiße Tips zur Zukunftssicherheit von Computern und Programmen und Informationen über neue Trends.
Samstag, 6 Februar, 12.00 Uhr, Saal 2
WEITERBILDUNGSZENTRUM am Hauptbahnhof, Bertha-von-Suttner-Platz 1, Saal 2, gebührenfrei

Kleines EDV-Museum Rolf Unterberg
Samstag, 6. Februar, 9.00 - 15.00 Uhr,
WEITERBILDUNGSZENTRUM am Hauptbahnhof, Bertha-von-Suttner-Platz 1, Saal 1, gebührenfrei

# Auftrag für Gelegenheitsanzeige

**Anschrift:**
**Roeske's Computer Mag.**
**c/o Roeske Verlag**
**Westring 59c**
**3440 Eschwege**

Unter der Rubrik „Kleinanzeigen" veröffentlichen wir Gelegenheitsanzeigen für Verkaufsangebote, Kauf- und Tauschgesuche, Kontaktaufnahme bzw. Erfahrungsaustausch usw.

Preise für „Kleinanzeigen":
Private Gelegenheitsanzeige je Druckzeile 5,— DM inkl. MwSt. Chiffregebühr je Anzeige 10,– DM.

Gewerbliche Gelegenheitsanzeige je Druckzeile 7,— DM inkl. MwSt. (dürfen nicht unter Chiffre erscheinen).

Name und Adresse _____

_____ Abo.-Nr. _____

_____ _____
Unterschrift Ich zahle sofort nach Rechnungserhalt. Datum

Ich wünsche folgenden Text zu veröffentlichen:

Zutreffendes ankreuzen

suche ☐ Software suche ☐ Hardware ☐ Tausch ☐ Versch.
biete an ☐ Software biete an ☐ Hardware ☐ Kontakte ☐ Chiffre

# Des »Cursus infinitus« erster Teil

**So, nun ist es endlich soweit - auch die vielen Benutzer des Commodore 64 werden in Zukunft nicht nur mit Programmen und Software-Reviews zu ihrem Recht kommen, sondern auch durch Rubriken wie Programmiertechnik, Tips und Tricks, usw.**

Lange haben wir uns mit der Frage beschäftigt, ob eine - die 64-Freaks mögen uns verzeihen - Kleincomputer-Rubrik in den Rahmen dieses Heftes paßt, bzw. überhaupt von Ihnen gewünscht wird.
Nun, Ihre Reaktionen auf die ersten beiden Hefte haben uns die Richtung gewiesen. Es gibt offensichtlich eine ganze Menge Einsteiger, die erst kürzlich zum C64 gefunden haben, und die deshalb auch noch etwas anderes von einem Computerheft erwarten als 'nur' Programme.
In Anbetracht der Tatsache, daß der C64 immer noch der meistverkaufte Heimcomputer - im ursprünglichen Sinne - ist, sowie seine Fähigkeiten für die neue Generation dieser Computer richtungsweisend gewirkt haben, entschlossen wir uns also , einen Teil jeder Ausgabe - und des Videos, eben diesem Rechner zu widmen.
Um die x-te Wiederholung eines eventuell schon jahrealten Kurses "Einführung in die Bedienung des C64" oder ähnliches zu vermeiden, legten wir für diese Serie ein neues, wie wir meinen sinnvolleres Konzept fest.
Im Verlauf dieses "Kurses" werden wir anhand der Programmierung eines professionellen Adventures versuchen, zunächst alle nichttrivialen Basic-Befehle des CBM-Basic 2.0 zu erläutern.
Gleichzeitig werden wir - Professionalität verlangt einen gewissen Maßstab - sowohl den Graphic-Chip als auch den Sound-Chip benutzen, um "das Outfit" unseres Programms zu gestalten, damit keine Wünsche offenbleiben.
Damit Sie in etwa wissen, was Sie in diesem Zusammenhang erwartet, hier eine kleine Liste der Themen (die keinen Anspruch auf Vollständigkeit erhebt, insbesondere sind wir für jede zusätzliche Anregung/Ergänzung dankbar), die wir zu behandeln gedenken.

**Programmiertechniken:**
Sowohl in Basic als auch in Assembler.

**Das Betriebssystem:**
Optimale Ausnutzung und Anpassung an bestimmte Problemstellungen.

**Graphik:**
Hochauflösungsmodus, Sprites, eigene Grphik-Routinen, Interrupt-Programmierung, Smooth- & Character-Scrolling, Animation, Pack- und Entpack-Routinen.

**Sound:**
Grundlagen, Interrupt-Programmierung.

Im Anschluß daran, nach der Fertigstellung des Programms also, werden wir für einige der wichtigsten höheren Programmiersprachen, die für den 64 verfügbar sind, eine jeweils in einer Ausgabe enthaltene Beschreibung mit einer Einführung in die wichtigsten Befehle und Eigenarten geben, so daß der aufmerksame Leser nach Abschluß dieses Kursteils ein ausbaufähiges und auf andere Rechner übertragbares Grundwissen, sowohl in Programmiertechnik, als auch einigen Programmiersprachen zur Verfügung haben wird.
Soweit zum software- bzw. programmspezifischen Teil unseres Kurses. Doch damit sind wir noch nicht am Ende.
Ausgehend vom speziellen Fall des Commodore 64 - der sich aufgrund der schon praktizierten extremen "Sezierung" seiner Innereien dazu sehr gut eignet - werden wir versuchen, Grundlagen der Rechnerorganisation zu erläutern.
Auf diesen, sowohl physikalischen als auch mathematischen Voraussetzungen, wollen wir Ihnen die technische Seite Ihres Computers, angefangen von einfachsten Schaltungen bis hin zur Chip-Architektur nahebringen.
Da die Entwicklung bis zu diesem Zeitpunkt den heutigen Stand der Technik (man denke nur an den Fortschritt bzgl. VLSI) schon wieder hinter sich gelassen haben dürfte, werden wir uns ständig mit den aktuellsten technischen Entwicklungen beschäftigen und uns bemühen, Sie anschaulich darüber zu informieren.
Im Augenblick ist dies jedoch noch Zukunftsmusik. Deshalb also: Willkommen zu unserem ersten Teil unseres »cursus infinitus«!

---

In diesem ersten Teil wollen wir uns hauptsächlich mit einem der wichtigsten Programmteile unseres Adventures beschäftigen, mit dem sogenannten "Parser". Neben der Darstellung wesentlicher Hilfsmittel der Stringverarbeitung eignet er sich auch recht gut als anschauliches Beispiel der Programmierung eines Suchvorgangs.
Was soll also unser Parser tun? Wie bekannt, erfolgt die Einflußnahme auf die meisten Adventures mit Ein- oder Zwei-Wort-Kommandos. Auch wir wollen uns auf ein oder zwei Wörter beschränken; die Praxis zeigt, daß längere Befehlsfolgen nur extrem selten benutzt werden.
Unser Parser hat also die Aufgabe, einen eingelesenen String (nennen wir ihn bf$, in Analogie zum Befehl, den er enthalten soll) zu "verarbeiten", nämlich zunächst zu überprüfen, ob er einen bekannten Befehl enthält. Desweiteren ist es notwendig, daß gleichzeitig eine Information darüber geliefert wird, welcher Befehl gegeben wurde.
Dazu müssen natürlich die möglichen Befehle dem Parser bekannt sein. Wir nehmen also an, in zwei FELDERN vb$( ) und nm$( ) seien die "zugelassenen" Verben und Nomen gespeichert - aus welchen Gründen eine Trennung dieser beiden sinnvoll ist, wird der Aufbau des Parsers zeigen.
Nun aber genug der Theorie, zunächst bitten wir Sie, das vollständige Unterprogramm, das bereits alle Bedingungen erfüllt, abzutippen.
Die Steuerzeichen lauten:
Zeile 120: Shift CLR
Zeile 130: Ctrl 8
Zeile 50240: Cursur left

---

Wie schon aus dem (mittlerweile abgespeicherten? nicht vergessen) vorliegenden Programmstück ersichtlich, werden wir das gesamte Programm "modular", das heißt aus kleinen übersichtlichen Einheiten zusammengesetzt, aufbauen.
Nebenbei ist dies neben einer ausführlichen Kommentierung ein wichtiges Kriterium für die Güte eines Programms, soll es doch nicht nur bestimmte Aufgaben erfüllen, sondern auch möglichst einfach und verständlich ge- und beschrieben sein, damit man auch nach Monaten noch nachvollziehen kann, was man programmiert hat, und daß man zumindestens Teile dieses Programms wiederverwenden kann.
In unserem Programm haben wir uns deshalb bemüht, die Kommentierung so zu gestalten, daß wir auf eine zusätzliche Programmbeschreibung verzichten können. Somit können wir uns auf die Darstellung einiger weniger Funktionen bzw. Techniken beschränken:

**Zur Unterprogrammtechnik:**
Der modulare Aufbau von Programmen erfordert entsprechende Elemente in der verwendeten Programmiersprache. Insbesondere sind Befehle notwendig, die es gestatten, häufig benutzte oder in sich abgeschlossene Programmteile getrennt vom eigentlichen Hauptprogramm zu schreiben und dann von dort aus aufzurufen.
Diese Befehle sind im Commodore-Basic **GOSUB** und **RETURN**.
Durch gosub Zeilennummer wird der Rechner veranlaßt, an die angegebene Stelle zu springen, wobei er sich, im Unterschied zu **GOTO** die Stelle "merkt", von der er abgesprungen ist.
Er arbeitet dann das angesprungene Unterprogramm ab und kehrt dann, sobald er auf ein return trifft, zu der Stelle, an der er auf das gosub stieß, zurück, macht also mit dem gosub folgenden Befehl weiter.
Nun können natürlich auch von Unterprogrammen aus andere Unterprogramme aufgerufen werden, so daß ein immer weiterer "Abstieg" in Bezug auf die "Hauptprogrammebene" erfolgt. Das CBM-Basic 2.0 läßt maximal einen zehnstufigen Abstieg zu - also 10 ineinander verschachtelte Unterprogramme.
Ein bis in die kleinste Einheit strukturiertes Programm besteht in seinem Kern nur noch aus vielen Unterprogrammaufrufen, so daß jede Funktion bis ins kleinste sichtbar wird.
So weit wollen wir es aber nicht treiben; Sie werden aber sehen, gerade in unserem Beispiel hat die Unterprogrammtechnik einige unübersehbare Vorteile.

**Spezielles zum Programm**
Wie gesagt, im Wesentlichen ist das Programm selbstdokumentierend: wir werden hier detailliert nur die "Suchtechnik" betrachten.
Ausgewertet werden nur die ersten drei Zeichen des eingegebenen Nomens bzw. Verbs, sie also sind relevant und kennzeichnen den Befehl.
Auf sie wird die Tabelle durchsucht, die der Übersichtlichkeit wegen aus "ausgeschriebenen" Wörtern aufgebaut ist. Hat man eine Übereinstimmung gefunden, so ordnet man dem Befehl die Position des "Tabellenwortes" zu, die also seinen Code darstellt.
Nun aber zu dem Trick, den wir noch zusätzlich angewendet haben: Oft genug kommt es vor, daß es für eine Tätigkeit oder einen Gegenstand mehrere gleichberechtigte Bezeichnungen gibt. Es ist somit sinnvoll, diesen Synonymen den gleichen Code zuzuordnen.
Dies erreichen wir folgendermaßen: Zu 'gehe' sind beispielsweise 'laufe', 'renne' in unserem Fall synonym. Dies kennzeichnen wir dadurch, daß wir sie in folgender Art und Weise in einer Tabelle ablegen:

```
vb$(n+4) = "nimm"
vb$(n+5) = "gehe"
vb$(n+6) = "laufe1"
vb$(n+7) = "renne2"
vb$(n+8) = "kletter"
vb$(n+9) = "suche"
```

Wie Sie sehen, wird an das Synonym quasi seine "Nummer" angehängt. Haben wir nun in unserem Suchverfahren festgestellt, daß der Befehl 'laufe' eingegeben wurde, so ergibt sich als Code nach unserer Formel (siehe Programm Zeile 51160 bzw. 51230): Code = Position - Nummer des Synonyms = n.
Dadurch erreichen wir, was gefordert war, nämlich einen gleichen Code aller bedeutungsgleichen Vokabeln.

Bevor wir den heutigen Teil beenden, wollen wir Ihnen noch eine Liste aller, in dieser Folge angesprochenen Speicherstellen, nebst anderer für die Bildschirmsteuerung interessanter Adressen notieren.

**197 = $C5**
Enthält den Tastaturcode (der ungleich allen anderen Codes ist!) der gerade gedrückten Taste. Ist keine solche gedrückt, gilt **PEEK(197)=64**. Diese Speicherstelle kann z.B. verwendet werden, um auf Tastendruck zu warten.Beispiel:

```
10 IF PEEK(197)=64 THEN 10.
```

**198 = $C6**
Enthält die Anzahl der Zeichen im Tastaturbuffer. Diese ist immer kleiner oder gleich zehn. POKE 198,0 löscht den Tastaturpuffer und kann somit zum Abfangen ungewünschter Eingaben benutzt werden.

**204 = $CC**
»Flag« (=Informationsbyte) für den Zusatnd des Cursors. Ist Peek (204)=0, so ist er sichtbar (z.B. beim Input-Befehl), ansonsten erscheint er nicht. Mit POKE 204,0 kann der Cursor z.B. für Eingaben mit GET sichtbar gemacht werden.

**297 = $CF**
»Flag« für die Blinkphase des Cursors. Ist PEEK(207)=0, so hat der Cursor die Hintergrundfarbe, ist faktisch unsichtbar. Ansonsten hat er die aktuelle Zeichenfarbe.
Ein Hinweis dazu: Wird der Cursor mit POKE 204,1 abgeschaltet, so sollte er zuerst mit POKE 207,0 unsichtbar gemacht werden, da er sonst eventuell auf dem Bildschirm sichtbar bleibt.

**211 = $D3**
Enthält die aktuelle Spaltenposition (x) des Cursors. X hat dabei einen Wert von 0 bis 39.

**214 = $D6**
Entspricht 211, enthält allerdings die Zeilenposition (y) des Cursors. Hier lautet der Wert 0 bis 24.
Ein Hinweis dazu:
POKE 211,X:POKE 214,Y:SYS58732 positioniert den Cursor an Zeile Y und Spalte X.

Alle diese Speicherstellen liegen im Bereich von 0 bis 255, der Seite Null bzw. Zeropage. Welche praktische Bedeutung dieser Bereich hat, werden wir im Kapitel über die Maschinenprogrammierung erläutern. Machen Sie sich ruhig mit den Funktionen der angesprochenen Speicherzellen vertraut. Probieren geht über Studieren. Kaputtgehen kann nichts!

Wenn auch dieser erste Teil im Wesentlichen von der Einführung in unseren "Cursus infinitus" bestimmt war, so hoffen wir doch, Ihnen klargemacht zu haben, wie wir uns diesen Kurs als ganzen vorstellen.
Sollten Sie dazu Anregungen, Kritik oder Verbesserungsvorschläge haben, so sind Sie herzlich eingeladen, sich damit an unserer Serie zu beteiligen - wir freuen uns über Reaktionen jeder Art.
In der nächsten Ausgabe werden wir unserem Programm die komplette Grundstruktur verleihen und in diesem Zusammenhang weitere grundlegende Programmiertechniken behandeln.

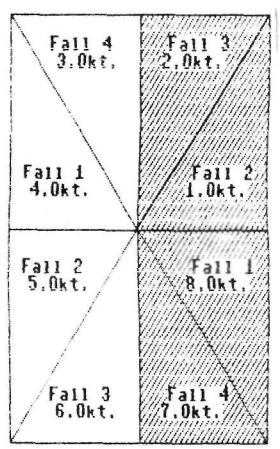
```
10 REM ********************************************************************************
11 REM **                                                                            **
12 REM **                        IN EINER STERNENKLAREN NACHT                        **
13 REM **                                 ODER:                                      **
14 REM **                       ELRONDS ABENTEUER IN MAGRATEA                        **
15 REM **                                                                            **
16 REM **                    DAS ADVENTURE ZUM "CURSUS INFINITUS"                    **
19 REM **                                                                            **
20 REM ********************************************************************************
21 :
100 NM%=11:VB%=6:DIMNM$(NM%),VB$(VB%) :REM ANZAHL DER VOKABELN (VORLAEUFIG)
105 GOSUB60000
110 REM VON 110-200 : NUR ZUM TESTEN NOTWENDIGER PROGRAMMTEIL
120 PRINT"■":POKE53280,0:POKE53281,0
130 PRINT"■EBEN SIE NUN IHRE BEFEHLE EIN. ES"
140 PRINT"WIRD DANN DER CODE VON NOMEN UND VERB"
150 PRINT"AUSGEGEBEN.":PRINT
```

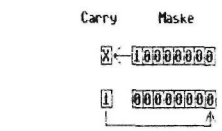
```
160 GOSUB51000                          :REM RUFE PARSER AUF
170 PRINT"CODE DES NOMEN :"B1%
180 PRINT"CODE DES VERB  :"B2%
190 PRINT:IFB1%<>NM%THEN160              :REM WIRD MIT "QUIT" BEENDET
200 END
300 :
400 :
500 :
600 :
700 :
50000 REM UNTERPROGRAMM BEFEHL EINLESEN ****************************************
50010 PRINT":";
50015 BF$=""                            :REM LOESCHE ALTEN BEFEHL
50020 POKE204,0                         :REM CURSOR EINSCHALTEN
50025 POKE198,0:WAIT198,1:GETC$         :REM WARTE AUF TASTENDRUCK
50030 REM UNTERSUCHE EINGELESENES ZEICHEN AUF GUELTIGKEIT
50035 C=ASC(C$)                         :REM ASCII-CODE DES ZEICHENS
50040 IFC=20THENGOSUB50200:GOTO50020    :REM DELETE-TASTE
50050 IFC=13THEN50150                   :REM RETURN=EINGABEENDE
50060 IFC<32THEN50020                   :REM ASCII-ZEICHEN BIS SPACE ABFANGEN
50070 IF(C>33)AND(C<65)THEN50020        :REM ZEICHEN VON " BIS ? ABFANGEN
50080 IFC>90THEN50020                   :REM GRAPHIK- UND SONDERZEICHEN ABFANGEN
50090 REM GUELTIGES ZEICHEN
50095 IFLEN(BF$)=20THEN50020            :REM LAENGE DER EINGABE BEGRENZEN
50100 BF$=BF$+C$                        :REM AN DEN BISHERIGEN BEFEHLSSTRING
50110 POKE207,0:POKE204,1               :REM CURSOR AUSSCHALTEN
50120 PRINTC$;                          :REM UND DANN ZEICHEN AUSGEBEN
50130 GOTO50020                         :REM JETZT WEITER EINLESEN
50150 REM BEFEHL MIT RETURN ABGESCHLOSSEN ***************************************
50160 IFLEFT$(BF$,1)=" "THENBF$=MID$(BF$,2):GOTO50160
50161 REM IN 50160 WERDEN EVENTUELL AM ANFANG STEHENDE LEERZEICHEN GELOESCHT
50170 POKE207,0:POKE204,1:PRINT" "
50180 IFLEN(BF$)=0THEN50000             :REM LEERE EINGABE, ALSO NEU EINLESEN
50190 RETURN                            :REM EINLESEN KORREKT BEENDET
50200 REM ZEICHEN AUS BF$ LOESCHEN *****************************************
50210 N=LEN(BF$)
50220 IFN=0THENRETURN                   :REM BF$ LEER ALSO NICHTS ZU LOESCHEN
50230 BF$=LEFT$(BF$,N-1)                :REM NEHME DIE N-1 ERSTEN ZEICHEN VON BF$
50240 POKE207,0:POKE204,1:PRINT" ■■■ ■■";:REM LOESCHE ZEICHEN AUF DEM BILDSCHIRM
50250 RETURN                            :REM DELETE BEENDET
50900 :
50910 :
50920 :
50930 :
50940 :
51000 REM UNTERPROGRAMM PARSER *************************************************
51005 B1%=0:B2%=0                       :REM CODES LOESCHEN
51010 GOSUB50000                        :REM LIES DEN BEFEHL EIN
51015 REM SUCHE NACH EINER TRENNENDEN LEERSTELLE IN BF$ ***********************
51020 I=1
51030 IFI=LEN(BF$)+1THENNM$=BF$:GOTO51100 :REM KEIN LEERZ., SETZE NOMEN=BEFEHL
51040 IFMID$(BF$,I,1)<>" "THENI=I+1:GOTO51030 :REM DURCHSUCHE BF$ NACH LEERZ.
51050 NM$=LEFT$(BF$,I-1)                :REM LINKER TEIL DES BEFEHLS = NOMEN
51060 VB$=MID$(BF$,I+1)                 :REM RECHTER TEIL DES BEFEHLS = VERB
51070 IFLEFT$(VB$,1)=" "THENVB$=MID$(VB$,2):GOTO51070
51071 REM IN 51070 WERDEN EVENTUELL AM ANFANG STEHENDE LEERZEICHEN GELOESCHT
51100 REM VORBEREITUNG ZUM SUCHEN NACH NOMEN UND VERB SIND ABGESCHLOSSEN, *****
51105 REM GESUCHT WIRD JEWEILS NUR NACH DEN ERSTEN DREI ZEICHEN VON NOMEN/VERB
51110 NM$=LEFT$(NM$,3):VB$=LEFT$(VB$,3):REM VERKUERZE ENTSPRECHEND
51120 REM SUCHE IN DER TABELLE *************************************************
51130 I%=1:N=LEN(NM$)
51140 IFI%=NM%+1THEN51300               :REM ENDE DER TABELLE, NICHT GEFUNDEN
51150 IFLEFT$(NM$(I%),3)<>NM$THENI%=I%+1:GOTO51140 :REM TABELLE DURCHSUCHEN
51160 B1%=I%-VAL(RIGHT$(NM$(I%),1))     :REM BEFEHLSCODE FUER DAS NOMEN
51200 I%=1:N=LEN(VB$):IFN=0THEN51250    :REM VERB EINGEGEBEN ? DANN SUCHEN
51210 IFI%=VB%+1THEN51300
51220 IFLEFT$(VB$(I%),3)<>VB$THENI%=I%+1:GOTO51240 :REM TABELLE DURCHSUCHEN
51230 B2%=I%-VAL(RIGHT$(VB$(I%),1))     :REM BEFEHLSCODE FUER DAS VERB
51250 RETURN                            :REM PARSING ERFOLGREICH BEENDET *******
51300 PRINT"???":PRINT:GOTO51000        :REM BEFEHL NICHT BEKANNT, NEUEINGABE
59900 :
59910 :
59920 :
59930 :
59940 :
60000 REM HIER FOLGT DIE INITIALISIERUNG DER TABELLEN **************************
60010 FORI=1TONM%:READNM$(I):NEXTI       :REM NOMEN EINLESEN
60020 FORI=1TOVB%:READVB$(I):NEXTI       :REM VERBEN EINLESEN
60090 RETURN                            :REM EINLESEN BEENDET
62999 REM HIER DIE NOMEN :  ************************************************
63000 DATANORDEN,OSTEN,SUEDEN,WESTEN,HOCH,RUNTER,TUER,PFORTE1,HAUS,HUETTE1
63400 DATAQUIT
63410 REM LETZES NOMEN: BEFEHL ZUM ABBRUCH
63499 REM HIER DIE VERBEN:  ************************************************
63500 DATAGEHEN,RENNEN1,LAUFEN2,NEHMEN,FASSEN1,GREIFEN2
63900 :
63910 :
63920 :
63930 :
63940 :
63950 REM LISTING ZUM ERSTEN TEIL DES "CURSUS INFINITUS" ********************
```

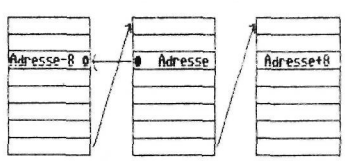
# Eine schnelle LINE-Routine

Über LINE wurde schon viel geschrieben und nachgedacht. Vor allem die Realisation auf Computern brachte schon manchen zum Verzweifeln.
Jeder, der einmal ein Grafikpaket oder ein Malprogramm entwikkeln wollte, stand einmal vor dem Problem, einen Algorithmus zum Verbinden zweier Punkte zu programmieren. Meist, sollte nach umständlichen Versuchen oder nach Entwickeln einer mathematischen Grundlage Erfolg beschieden sein, war doch noch ein Haken an der Sache: Die Geschwindigkeit ließ zu wünschen übrig.
Dem soll jetzt abgeholfen werden. Entgegen vielleicht anders lautenden Behauptungen ist dies wirklich eine der schnellsten Line-Algorithmen, die es für den C64 (bisher) gibt.
Mit über 30 000 Punkten pro Sekunde werden hier die Linien im günstigsten Fall gezeichnet. Allerdings muß hier gesagt werden, daß sich je nach Lage der Linie eine erhebliche Geschwindigkeitsminderung ergibt (bis auf ca 12 000 Punkte pro Sekunde). Das gibt aber ein gutes Mittel für das Zeichnen einer normalen Linie ( ca 20 000 Punkte pro Sek.).
Zur Geschwindigkeitssteigerung werden eine Reihe Programmier- und Programmtricks angewandt. So wird z.B. eine Linie nicht mit einem Punkt gezeichnet, sondern mit zweien, die immer in die gegensätzliche Richtung bewegt werden. Weiterhin wird das Feld um den Startpunkt in (Acht) Oktanten zerlegt. Vier davon entfallen, da diese Fälle in den anderen schon enthalten sind und einfach durch Vertauschen der Koordinaten erreicht werden können. So braucht nur noch zwischen vier Fällen unterschieden zu werden.
Diese wurden als einzelne Programmteile programmiert. Damit entfällt das lästige und zeitaufwendige Entscheiden der Richtung für einen Punkt.
Nicht zuletzt wurde noch so oft wie möglich die Immediate-Adressierungsart verwendet, welche meist einen Taktzyklus schneller ist, als die Zero-Page Adressierung.
Weitere Programmveränderungen, wie Adressenänderungen hinter Load-and-Store Befehlen, wurden nicht ausgelassen.
Erwähnenswert wäre noch, daß dieses Drawline-Programm nur x-Koordinaten von 0 bis 255 zuläßt und damit den 'Zeichenbereich' etwas einschränkt, was aber einfach zu ändern ist und erfahrenen Programmierern keine Schwierigkeiten bereiten dürfte.

Zum Programm:
Im Programm wird grundsätzlich von rechts nach links (Bezogen auf Punkt 1) gezeichnet. Punkt 2 macht also für jeden Schritt, den Punkt 1 nach links macht, einen Schritt nach rechts. Jede Seitwärtsbewegung ist also eine rechts-links Bewegung.
In Bild 1 ist zu erkennen, wie die einzelnen Fälle auf die Oktanten aufgeteilt sind. In Fall 1 und 2 wird pro Schleifendurchlauf auf jeden Fall ein Schritt in der Horizontalen durchgeführt. Es muß also nur noch entschieden werden, ob ein Schritt in der Senkrechten nötig ist. Das wird anhand des Approximationswertes geprüft.
Dieser enthält zu Beginn die Hälfte der Maximaldifferenz der Koordinaten. Das bedeutet (auf Fall 1 und 2 bezogen) die Hälfte der x-Differenz, da diese größer ist als die y-Differenz.
Zu diesem Approximierungswert (Näherungswert) wird die kleinere Differenz (hier die y-Differenz) addiert. Solange das Ergebnis kleiner der Maximaldifferenz (x-Differenz) ist, wird ein seitlicher Schritt durchgeführt. Wenn der Wert größer ist, wird die Maximaldifferenz wieder abgezogen, damit der Approximierungswert für nachfolgende Berechnungen wieder in den normalen Bereich gerät.
Das Gleiche muß auch getan werden, wenn durch die Addition das Ergebnis größer als 255 geworden ist. Im Prinzip ist das eben beschriebene Verfahren, obwohl es nur mit Addition und Subtraktion arbeitet, nichts anderes, als eine auf die Grundstruktur zurückgeführte Division, denn ganz ohne kommt man kaum aus. Es gibt nur verschiedene Verfahren, sie zu programmieren.
Zu den Fällen 3 und 4 ist eigentlich nur zu sagen, daß dort die Achsen vertauscht werden und statt waagerechten Schritten, senkrechte und statt senkrechten, waagerechte Schritte ausgeführt werden. Damit sind alle vier Fälle erledigt.
Bleibt nur noch zu sagen, daß das Programm speziell auf den Grafikspeicher des C64 abgestimmt wurde. Es wird in den Hauptschleifen nicht mehr zu 'Plot' gesprungen, sondern nur zum Beginn der Schleife. Dort werden dann die Anfangsadressen der beiden Punkte und deren Masken (Positionen im Byte) berechnet und abgespeichert.
In der Schleife wird z.B. bei einem Linksschritt nur die Maske um eine Position nach links verschoben. Aufpassen muß man nur, wenn das 1-Bit aus der Maske geschoben wird. Das bedeutet, daß der zuletzt geplottete Punkt am linken Rand eines 8er-Päckchens (siehe Bild 2) gelegen hat. Die Maske muß natürlich wieder korrigiert werden (das einzige Bit wurde ja links herausgeschoben), was man durch Hereinschieben auf der rechten Seite erreicht. Nun muß nur noch die Adresse um 8 verringert werden, da man ja im linken Nachbarpäckchen weiterzeichnen will.
Der Schritt nach rechts verläuft ganz analog, nur daß hier 8 addiert werden muß, um in das rechte Nachbarpäckchen zu gelangen.
Bei den senkrechten Schritten muß man ein wenig weiter ausholen. Der Grafikschirm besteht aus 25 Zeilen mit je 320 Bytes, welche wieder zu je 8 gebündelt sind. Von Zeile zu Zeile besteht also ein Offset von 320. Innerhalb eines 8er-Päckchens braucht beim Schritt nach oben nur die Adresse um eins verringert zu werden, weil dort der Abstand der Bytes nur halt den Wert 1 hat. Befindet sich aber der letzt gezeichnete Punkt am oberen Rand der Grafikzeile, so muß ein Sprung von 320 Bytes gemacht werden, um an die gleiche Position in der darüberliegenden Zeile zu kommen. Da man aber in der darüberliegenden Zeile nicht den oberen, sondern den unteren Rand erreichen will, muß also nicht nur 320 von der Adresse abgezogen werden, sondern auch ein Wert von 7 addiert werden, damit der untere Rand der Grafikzeile erreicht wird. Das macht zusammen eine Subtraktion von 313, wie sie auch im Programm steht.
Man kann sagen, daß alle Tabellen, die im Programm verwendet werden nur dazu dienen, nötige Berechnungen auf ein Mindestmaß zu reduzieren, um so die Ausführungszeit weiter zu senken. Bevor sie nun den Line-Algorithmus aufrufen, müssen die Tabellen initialisiert werden. Das geschieht mit SYS 7*4096+3.



```
5 rem clr:open1,8,15,"s:fastline":close1
:save"fastline",8
10 poke56,112
20 clr
100 sys2^15
110 .opt oo
120 .pag 71
130 .tit "fastline j.s. 9/86"
140 .sym
150 *= $7000
160 ptr1 = $f7
170 ptr2 = $f9
180 ptr4 = $fd
190 xk   = $ac
200 yk   = $ae
210 xk1  = $bb
220 yk1  = $bd
230 sto0 = $a6
240 sto1 = $8e
250 msk  = $b6
260 mskk = $07
270 xtab  = $7c00
280 msk1  = $7d00
290 ptabl = $7e00
300 ptabh = $7f00
310 ;
320 chkcom = $aefd
330 getbyt = $b79e
340 ;
350 vic = $d000
360 ;
1000 jmp line;sprung zur line-routine
1010 ;
1020 lda #8*16;einschalten der grafik
1030 sta vic+24;bei $4000.
1040 lda #59;grafik ein
1050 sta vic+17
1060 lda #$02;2. 16kbyte abschnitt des
1070 sta $dd00;videochips ansteuern
1080 ;
1090 lda #<$4000;grafik loeschen
1100 ldy #>$4000;dazu pointer mit
1110 sta ptr1;grafikstart laden
1120 sty ptr1+1
1130 tay;0 in das y-register
1140 ldx #$20;32 pages loeschen
1150 cl1 sta (ptr1),y;null byte setzen
1160 iny;low-byte erhoehen
1170 bne cl1;kein uebertrag,dann cl1
1180 inc ptr1+1;high-byte erhoehen
1190 dex;seitenzahl erniedrigen
1200 bne cl1;"ungleich null ? dann cl1
1210 lda #$10;farbe (1*16+0) weiss auf s
chwarz
1220 cl2 sta $6000,y;im farbspeicher
1230 sta $6100,y;zur grafik ablegen
1240 sta $6200,y
1250 sta $6300,y
1260 iny;low-byte erhoehen
1270 bne cl2;"kein uebertrag ? dann cl2
1280 jmp lineinit;tabelle aufstellen
1290 ;
1300 line sei;interrupt ausschalten
1310 jsr chkcom;im basic-text komma such
en
1320 jsr getbyt;bytewert (0-255) holen
1330 stx xk;ins x-register,das in xk spe
ichern
1340 jsr chkcom;komma-abfrage
1350 jsr getbyt;bytewert holen
1360 stx yk;x-register nach yk speichern
1370 jsr chkcom;komma holen
1380 jsr getbyt;bytewert holen
1390 stx xk1;x-register nach xk1
1400 jsr chkcom;komma-abfrage
1410 jsr getbyt;bytewert holen
1420 stx yk1;nach yk1
1430 ;
1440 jsr line1;einsprung line
1450 cli;interrupt wieder freigeben
1460 rts;ruecksprung zum basic
1470 ;
1480 ;
1490 ;aufstellen einer tabelle mit den a
nfangsadressen aller
1500 ;grafikzeilen.da eine grafik 200 ze
ilen hat und zeiger auf
1510 ;grafikteile 2 bytes lang sind, erg
ibt sich eine tabellen-
1520 ;laenge von 200*2 = 400 bytes.
1530 ;aus geschwindigkeitsgruenden wurde
n hi- und low-bytes
1540 ;der zeiger in getrennten 256 byte-
pages untergebracht.
1550 ;(sonst muss beim zugriff 1 taktzyk
lus wegen seitenuber-
1560 ;schreitung dazuaddiert werden)
1570 ;
1580 ;eine weitere tabelle dient dazu, d
ie masken, die angeben,
1590 ;welches bit in die grafik gesetzt
wird, zu speichern.
1600 ;
1610 ;die letzte tabelle gibt an, wann e
in indexregister, mit
1620 ;"dem man auf diese tabelle zugreif
t, der bedingung
1630 ;(register and 7)=7 genuegt. in die
sem falle ist der wert
1640 ;in der tabelle = 0, ansonsten 255.
1650 ;
2000 lineinit lda #<$4000;adresse der
2010 ldy #>$4000;1.grafikzeile
2020 sta ptr1
2030 sty ptr1+1
2040 tax;x-reg. = 0
2050 tay;y-reg. =0
2060 clc;carry fuer addition loeschen
2070 l1 tya;offset zwischen 0 und 7
2080 adc ptr1;zum pointer addieren
2090 sta ptabl,x;und abspeichern
2100 lda ptr1+1;uebertrag beruecksich-
2110 adc #$00;tigen
2120 sta ptabh,x
2130 inx;anzahl der grafikzeilen erhoehe
n
2140 iny;offset erhoehen
2150 cpy #$08;"kleiner als 8 ?
2160 bcc l1;ja, dann l1
2170 lda ptr1;nein, carry gesetzt
2180 adc #<319;319 + carry (=320) zum
2190 sta ptr1;pointer addieren
2200 lda ptr1+1;high-byte berucksichtige
n
2210 adc #>319
2220 sta ptr1+1
2230 ldy #$00;paeckchenoffset wieder 0
2240 cpx #$c8;"anzahl zeilen < 200 ?
2250 bcc l1;ja, ==> schleife
2260 ;
2270 l2 lda #$80;akku = 128 (%10000000)
2280 ldx #$08;8 bitpositionen
2290 l3 sta msk1,y;akku in maskentabelle
 schreiben
2300 lsr;bit einen schritt nach rechts
2310 iny;tabellenzeiger erhoehen
2320 dex;anzahl positionen erniedrigen
2330 bne l3;wenn nicht 0,dann l3
2340 tya;"tabellenende ?
2350 bne l2;nein, ==> l2
2360 ;
2370 l4 lda #$ff;wert ungleich null (hie
r 255)
2380 ldx #$07;7 mal schreiben
2390 l5 sta xtab,y;wert in tabelle schre
iben
2400 iny;tabellenposition erhoehen
2410 dex;anzahl der schreibzugriffe - 1
2420 bne l5;"schon 0 ?
2430 txa;ja, null in akku
2440 sta xtab,y;und 0-byte schreiben
2450 iny;position in tabelle erhoehen
2460 bne l4;"noch nicht fertig ? dann ==
> l4
2470 rts;fertig mit initialisierung
2480 ;
```

```
2490 ;-----------------------------------
-----------------------
2500 ;
2510 ;richtungsangaben im programm bezie
hen sich immer
2520 ;auf die plotrichtung von punkt 1 !
!
2530 ;
2540 ;-----------------------------------
-----------------------
2550 ;
3000 line1 sec;carry fuer subtraktion se
tzen
3010 lda xk;akku = xk1 - xk
3020 sbc xk1
3030 bcs line2;c=1 wenn xk1 < xk
3040 ;
3050 ldx xk;koordinaten vertauschen
3060 ldy xk1;hier x-koordinaten
3070 stx xk1
3080 sty xk
3090 ldx yk;hier y-koordinaten
3100 ldy yk1;damit wird die richtung der
3110 stx yk1;linie nicht veraendert,sond
ern
3120 sty yk;nur der 'anfangspunkt' ausge
tauscht
3130 ;(es wird immer von rechts nach lin
ks gezeichnet)
3140 eor #$ff;2-er komplement (akku=-akk
u)
3150 adc #$01;bilden (carry=0 !)
3160 ;
3170 sec;carry fuer subtraktion setzen
3180 line2 sta sto0;x-differenz in sto0
speichern
3190 ;
3200 ldy #$00;y-register fuer plot mit 0
 laden
3210 lda yk;akku = yk - yk1
3220 sbc yk1
3230 bcs line4;yk1 < yk ==> line4
3240 ;von oben nach unten zeichen !!
3250 eor #$ff;2er komplement bilden
3260 adc #$01
3270 sta sto1;differenz nach sto1
3280 cmp sto0;mit x-differenz vergleiche
n
3290 bcc line3
3300 lsr;maximaldifferenz (y-diff) halbi
eren
3310 sta ptr4;=anzahl doppelt zu setzend
er punkte
3320 sta ptr4+1;= 1. approximationswert
3330 jmp linec;6.oktant
3340 ;
3350 line3 lda sto0;xdiff ( hier groesst
e differenz )
3360 lsr;halbieren als
3370 sta ptr4;anzahl der doppelpunkte un
d
3380 sta ptr4+1;1. approximationswert sp
eichern
3390 jmp lineb;5.oktant
3400 ;
3410 line4 sta sto1;ydiff abspeichern
3420 ;
3430 lda sto0;xdiff mit ydiff vergleiche
n
3440 cmp sto1;"xdiff>=ydiff ?
3450 bcs line5;ja
3460 ldx sto1;xdiff mit ydiff vertausche
n
3470 sta sto1
3480 stx sto0
3490 txa
3500 lsr;maximaldifferenz halbieren
3510 sta ptr4;anzahl doppelpunkte
3520 sta ptr4+1;1. approximationswert
3530 jmp lined;3.oktant
3540 ;
3550 line5 lsr;wie oben
3560 sta ptr4
3570 sta ptr4+1;4.oktant
3580 ;
3590 ;-----------------------------------
4000 ;
4010 ; fall 1 (4.oktant)
4020 ;
4030 linea jsr plot;ersten zwei punkte s
etzen
4040 ;
4050 lda ptr4;"nicht mehr punkte ?
4060 bne linea0;doch
4070 rts;sonst line beendet
4080 ;
4090 linea0 lda sto0;aus geschwindigkeit
s-
4100 sta ssto0a+1;gruenden wird nun eine
4110 sta linea1+1;programmveraenderung
4120 lda sto1;durchgefuehrt. die immedia
te
4130 sta ssto1a+1;adressierungsart ist
4140 ;schneller als zeropage-adressierun
g !
4150 ;
4160 ldy ptr1;die low-bytes der adressen
4170 ldx ptr2;der 2 punkte in die index-
register
4180 ;
4190 lda ptr1+1;die high-bytes in den
4200 sta linea8+2;programmtext !
4210 sta linea9+2;(aus gruenden der
4220 lda ptr2+1;geschwindigkeitssteigeru
ng)
4230 sta linea10+2
4240 sta linea11+2
4250 ;
4260 loopa lda ptr4+1;zum approximations
wert
4270 ssto1a adc #sto1;ydiff addieren
4280 bcs linea1;">255 ? dann ==> linea1
4290 sta ptr4+1;sonst als neuen appr.wer
t
4300 ;
4310 ssto0a cmp #sto0;und mit xdiff verg
l.
4320 bcc linea6;kleiner, dann ok, schrit
t links
4330 ;
4340 linea1 sbc #sto0;sonst ydiff abzieh
en
4350 sta ptr4+1;(normalisierung von appr
.wert)
4360 ;1.punkt schritt nach oben
4370 lda msk1,y;"(yreg. and 7)=0 ?
4380 ;(abfrage oberer paeckchenrand)
4390 bpl linea3;nein, ==> linea3
4400 sec;ja, carry fuer subtraktion
4410 tya;low-byte adresse 1.punkt
4420 sbc #<313;313 abziehen,um in hoeher
-
4430 tay;gelegenes paeckchen zu gelangen
4440 lda linea8+2;hi-byte aus programmte
xt
4450 sbc #>313;int(313/256) abziehen unt
er be-
4460 sta linea8+2;ruecksichtigung des ca
rrys
4470 sta linea9+2;in das programm schrei
ben
4480 .byt $24;naechsten befehl uebersprin
gen
4490 ;
4500 linea3 dey;nur low-byte erniedrigen
4510 ;2.punkt schritt nach unten
4520 linea4 lda xtab,x;"(xreg. and 7)=7
?
4530 ;"unterer paeckchenrand ?
4540 bne linea5;nein,nur low-byte erhoeh
en
4550 ;
4560 clc;carry fuer addition loeschen
4570 txa;(xreg.=low-byte der adresse 2.p
unkt)
4580 adc #<313;313 addieren
```

```
4590 tax;und wieder abspeichern
4600 lda linea10+2;hi-byte aus programm
4610 adc #>313;int(313/256) addieren (+c
arry)
4620 sta linea10+2;wieder in das program
m
4630 sta linea11+2;hinter den sta-befehl
4640 .byt $24;naechsten befehl (1byte) u
eberspr.
4650 ;
4660 linea5 inx;nur low-byte erhoehen
4670 ;"1.punkt schritt links
4680 linea6 asl msk;"plot-maske links ve
rschieben
4690 bcc linea7;"bit herausgeschoben ?
4700 rol msk;ja,auf der anderen seite wi
der rein
4710 tya;"anfangsadresse ins linke nachb
ar-
4720 sbc #$07;paeckchen verlegen (carry=
0)
4730 tay
4740 bcs linea7;"unterlauf ?
4750 dec linea8+2;ja, high-bytes auch er
-
4760 dec linea9+2;niedrigen
4770 ;2.punkt schritt rechts
4780 linea7 lsr mskk;maske rechts versch
ieben
4790 bcc linea8;"noch innerhalb byte ?
4800 ror mskk;nein,korrigieren
4810 txa;ins rechte nachbarpaeckchen
4820 adc #$08;zur adresse 8 addieren
4830 tax
4840 bcc linea8;"uebertrag ?
4850 inc linea10+2;ja, high-bytes erhoeh
en
4860 inc linea11+2
4870 clc;carry wieder loeschen
4880 ;
4890 ;beide punkte plotten
4900 ;
4910 linea8 lda $ff00,y;die adressen hin
ter diesen
4920 ora msk;befehlen werden vom program
m
4930 linea9 sta $ff00,y;geaendert
4940 linea10 lda $ff00,x;zweiten punkt s
etzen
4950 ora mskk
4960 linea11 sta $ff00,x
4970 ;
4980 dec ptr4;anzahl der zu setzenden pu
nkte-1
4990 bne loopa;"<>0 ? dann schleife
5000 ;
5010 rts;minimal 64 takte fuer 2 punkte
5020 ;im guenstigsten fall also 30789 p/
sek
5030 ;---------------------------------
6000 ;
6010 ; fall 2 (5.oktant)
6020 ;
6030 lineb jsr plot;anfangspunkte setzen
6040 ;adressen und masken der punkte erm
itteln
6050 lda ptr4;"anzahl der punkte < 2 ?
6060 bne lineb0;nein ===> lineb0
6070 rts;line beendet
6090 ;
6100 lineb0 lda sto0;xdiff hinter befehl
6110 sta ssto0b+1;schreiben
6120 sta lineb1+1
6130 lda sto1;das gleiche mit ydiff
6140 sta ssto1b+1
6150 ;
6160 ldy ptr1;zur schnelleren handhabung
 der
6170 ldx ptr2;low-bytes diese in x und y
6180 ;
6190 lda ptr1+1;high-bytes ins programm
6200 sta lineb8+2;1.punkt
6210 sta lineb9+2
6220 lda ptr2+1;2.punkt
6230 sta lineb10+2
6240 sta lineb11+2
6250 ;
6260 loopb lda ptr4+1;zum appr.wert
6270 ssto1b adc #sto1;ydiff addieren
6280 bcs lineb1;uebertrag beachten
6290 sta ptr4+1;als neuen startwert spei
chern
6300 ;
6310 ssto0b cmp #sto0;vergleichen mit xd
iff
6320 bcc lineb6;wenn kleiner dann haupts
chleife
6330 ;
6340 lineb1 sbc #sto0;sonst korrigieren
6350 sta ptr4+1;(xdiff abziehen)
6360 ;1.punkt schritt nach unten
6370 lda xtab,y;unterer packenrand errei
cht
6380 bne lineb3;nein, low-byte erhoehen
6390 clc;carry fuer addition loeschen
6400 tya;low-byte um (313 and 255)
6410 adc #<313;erhohen
6420 tay;(oberer paeckchenrand)
6430 lda lineb8+2;hi-bytes um int(313/25
6)
6440 adc #>313;erhoehen
6450 sta lineb8+2;programmadressen
6460 sta lineb9+2;aendern
6470 .byt $24;iny-befehl (1 byte) uebers
pringen
6480 ;
6490 lineb3 iny;innerhalb eines paeckche
ns lb+1
6500 ;2.punkt schritt nach oben
6510 lineb4 lda msk1,x;"oberer paeckchen
rand ?
6520 bpl lineb5;nein,nur lb erniedrigen
6530 ;
6540 sec;carry fuer subtraktion setzen
6550 txa;zum unteren rand des oberen
6560 sbc #<313;paeckchens
6570 tax;low-byte
6580 lda lineb10+2;high-byte beruecksich
tigen
6590 sbc #>313
6600 sta lineb10+2;adressen korrigieren
6610 sta lineb11+2
6620 .byt $24;dex ueberspringen
6630 ;
6640 lineb5 dex;innerhalb paeckechen nur
 lb - 1
6650 ;"1.punkt schritt links
6660 lineb6 asl msk;"innerhalb byte ?
6670 bcc lineb7;ja, ==> lineb7
6680 rol msk;nein, korrektur
6690 tya;"linkes nachbarpaeckchen
6700 sbc #$07;ansteuern
6710 tay
6720 bcs lineb7;"unterlauf ?
6730 dec lineb8+2;ja, high-bytes korrigi
eren
6740 dec lineb9+2
6750 ;2.punkt schritt rechts
6760 lineb7 lsr mskk;maske nach rechts
6770 bcc lineb8;"noch innerhalb des paec
kchens ?
6780 ror mskk;nein,carry 'hereinrollen'
6790 txa;ins rechte nachbarpaeckchen
6800 adc #$08;wechseln
6810 tax;low-byte ins x-reg.
6820 bcc lineb8;"uebertrag ?
6830 inc lineb10+2;ja, high-byte erhoehe
n
6840 inc lineb11+2
6850 clc;carry fuer spaetere add. loesch
en
6860 ;
6870 ;plotten der beiden punkte
6880 ;1.punkt
6890 lineb8 lda $ff00,y;byte laden
6900 ora msk;entsprechendes bit setzen
```

```
6910 lineb9 sta $ff00,y;und wieder abspe
ichern
6920 ;2.punkt
6930 lineb10 lda $ff00,x;byte aus grafik
6940 ora mskk;mit 2.maske odern
6950 lineb11 sta $ff00,x;wieder in grafi
k zurueck
6960 ;
6970 dec ptr4;anzahl punkte -1
6980 bne loopb;"alle punkte gesetzt ?
6990 ;
7000 rts;ja,ende (minimal 64 takte)
7010 ;geschwindigkeitsmaximalfall 30789
p/sek
7020 ;---------------------------------
8000 ;
8010 ; fall 3 (6.oktant)
8020 ;
8030 linec jsr plot;2 punkte setzen
8040 ;
8050 lda ptr4;"keine punkte mehr ?
8060 bne linec0;nein, ==> linec0
8070 rts;ansonsten ende
8080 ;
8090 linec0 lda sto1;ydiff in's programm
8100 sta ssto1c+1;'poken'
8110 sta linec1+1
8120 lda sto0;xdiff auch
8130 sta ssto0c+1
8140 ;
8150 ldy ptr1;low-bytes der punktadresse
n
8160 ldx ptr2;in indexregister
8170 ;
8180 lda ptr1+1;high-bytes der punkt-
8190 sta linec8+2;adressen speichern
8200 sta linec9+2;(1.punkt)
8210 lda ptr2+1;(2.punkt)
8220 sta linec10+2
8230 sta linec11+2
8240 ;
8250 loopc lda ptr4+1;approximationswert
8260 ssto0c adc #sto0;+ xdiff
8270 bcs linec1;">256 ? ja, ==> linec1
8280 sta ptr4+1;nein,speichern
8290 ;
8300 ssto1c cmp #sto1;mit ydiff vergleic
hen
8310 bcc linec4;wenn kleiner, dann nur s
enkrecht
8320 ;
8330 linec1 sbc #sto1;carry = 1 !!
8340 sta ptr4+1;korrigierter appr.wert
8350 ;"1.punkt schritt links
8360 linec2 asl msk;"bit nach links
8370 bcc linec3;innerhalb, dann ==> line
c3
8380 rol msk;ausserhalb,wird aber berich
tigt
8390 tya;yreg-#$07 -(1-carry) { carry=0
}
8400 sbc #$07
8410 tay
8420 bcs linec3;kein unterlauf,dann line
c3
8430 dec linec8+2;high-bytes -1
8440 dec linec9+2
8450 ;2.punkt schritt rechts
8460 linec3 lsr mskk;maske nach rechts
8470 bcc linec4;"bit nach draussen ?
8480 ror mskk;wieder rein
8490 txa;adresse auf rechtes nachbarpaec
kchen
8500 adc #$08;setzen
8510 tax
8520 bcc linec4;"mit uebertrag ?
8530 inc linec10+2;ja, high-bytes
8540 inc linec11+2;erhoehen
8550 clc;fuer addition in zeile 8600
8560 ;1.punkt schritt runter
8570 linec4 lda xtab,y;"(yreg and 7)=7 ?
8580 bne linec5;nein,innerhalb packen
8590 tya;darunter liegenden packen
8600 adc #<313;bearbeiten
8610 tay
8620 lda linec8+2;high-bytes bearb.
8630 adc #>313
8640 sta linec8+2;fuer lda- und
8650 sta linec9+2;fuer sta- befehl
8660 .byt $24;mit bit-befehl iny uebersp
r.
8670 ;
8680 linec5 iny;schritt nach unten inner
halb
8690 ;2.punkt schritt hoch
8700 linec6 lda msk1,x;"(xreg. and 7)=0
?
8710 bpl linec7;nein (noch innerhalb pac
ken)
8720 ;
8730 sec;ja, also in darueberliegendes
8740 txa;paeckchen wechseln
8750 sbc #<313;adresse=adresse-313
8760 tax
8770 lda linec10+2;high-byte
8780 sbc #>313
8790 sta linec10+2
8800 sta linec11+2
8810 clc;fuer add. in zeile 8260
8820 .byt $24;dex ueberspringen
8830 ;
8840 linec7 dex;1b erniedrigen
8850 ;
8860 ;punkte plotten
8870 ;1.punkt
8880 linec8 lda $ff00,y;byte laden
8890 ora msk;mit maske verknuepfen
8900 linec9 sta $ff00,y;und speichern
8910 ;2.punkt
8920 linec10 lda $ff00,x;das gleiche
8930 ora mskk;mit dem 2. punkt
8940 linec11 sta $ff00,x
8950 ;
8960 dec ptr4;anzahl der punkte-1
8970 bne loopc;"noch nicht alle ? dann s
chleife
8980 ;                    ;---------
8990 rts;insgesammt minimal 65 takte
9000 ;entspricht maximal 30315 p/sek
9010 ;---------------------------------
10000 ;
10010 ; fall 4 (3.oktant)
10020 ;
10030 lined jsr plot;2 punkte setzen
10040 ;adressen und masken berechnen
10050 lda ptr4;anzahl der punkte
10060 bne lined0;<>0 print dann weiter
10070 rts;sonst ende
10080 ;nicht verwirren lassen ! sto0 und
 sto1 sind vertauscht
10100 lined0 lda sto0;ydiff
10110 sta ssto0d+1;ins programm schreibe
n
10120 sta lined1+1
10130 lda sto1;xdiff
10140 sta ssto1d+1;auch abspeichern
10150 ;
10160 ldy ptr1;low-bytes wie gehabt in
10170 ldx ptr2;index-register aufnehmen
10180 ;
10190 lda ptr1+1;high-bytes hinter
10200 sta lined8+2;lda-
10210 sta lined9+2;und sta-befehle
10220 lda ptr2+1;schreiben
10230 sta lined10+2;auch fuer 2. punkt
10240 sta lined11+2
10250 ;
10260 loopd lda ptr4+1;approximierungswe
rt
10270 ssto1d adc #sto1;plus xdiff
10280 bcs lined1;"uebertrag ?
10290 sta ptr4+1;nein, also abspeichern
10300 ;
10310 ssto0d cmp #sto0;und mit ydiff ver
gleichen
10320 bcc lined4;kleiner,dann ok senkrec
```

```
hte bewegung
10330 ;
10340 lined1 sbc #sto0;sonst ydiff abzie
hen
10350 sta ptr4+1;damit korrekter neuer a
ppr.wert
10355 ;entsteht
10360 ;"1.punkt schritt links
10370 lined2 asl msk;"maske links
10380 bcc lined3;"innerhalb byte ?
10390 rol msk;nein, korrigieren
10400 tya;adresse=adresse-8
10410 sbc #$07;(subtraktion mit geloesch
tem carry)
10420 tay
10430 bcs lined3;"mit unterlauf ?
10440 dec lined8+2;ja, beruecksichtigen
10450 dec lined9+2;(adresse hinter lda u
nd sta)
10460 ;2.punkt schritt rechts
10470 lined3 lsr mskk;maske verschieben
10480 bcc lined4;test ob noch innerhalb
10490 ror mskk;korrigierung
10500 txa;adresse=adresse+8
10510 adc #$08;carry auf jeden fall gelo
escht
10520 tax
10530 bcc lined4;"mit uebertrag ?
10540 inc lined10+2;ja, dann high-bytes
10550 inc lined11+2;aendern
10560 clc;carry fuer naechste add. loesc
hen
10570 ;1.punkt schritt nach oben
10580 lined4 lda msk1,y;"innerhalb packe
n ?
10590 bpl lined5;ja, lb=lb-1
10600 tya;adresse=adresse-313
10610 sbc #<312;(geloeschtes carry !)
10620 tay
10630 lda lined8+2;high-byte bearb.
10640 sbc #>312
10650 sta lined8+2
10660 sta lined9+2
10670 clc;zur sicherheit fuer zeile 1027
0
10680 .byt $24;1 byte ueberspringen
10690 ;
10700 lined5 dey;adresse=adresse-1
10710 ;
10720 lined6 lda xtab,x;"(xreg. and 7)=7
 ?
10730 bne lined7;nein,es genuegt erhoehu
ng des lb
10740 ;
10750 clc;grafikcursor schon an unterer
10760 txa;zeile des packens, also
10770 adc #<313;wechsel in die oberste
10780 tax;zeile des darunter liegenden
10790 lda lined10+2;packens (adresse+313
)
10800 adc #>313
10810 sta lined10+2
10820 sta lined11+2
10830 .byt $24;ueberspringe 1-byte befehl
10840 ;
10850 lined7 inx;erhoehe adresse um 1
10860 ;
10870 ;plotten der beiden punkte
10880 ;1.punkt
10890 lined8 lda $ff00,y;byte holen
10900 ora msk;mit maske verknuepfen
10910 lined9 sta $ff00,y;und abspeichern
10920 ;2.punkt
10930 lined10 lda $ff00,x;byte laden
10940 ora mskk;verknuepfen
10950 lined11 sta $ff00,x;abspeichern
10960 ;
10970 dec ptr4;anzahl doppelpunkte-1
10980 bne loopd;wenn nicht fertig >> loo
pd
10990 ;
11000 rts;minimal 65 takte
11010 ;entspricht 30315 punkten pro seku
nde
50040 ;
50050 ;es wird erwartet dass sich im y-r
egister
50060 ;"der wert null befindet.
50070 ;
60000 plot clc
60010 ldx yk;zeile ins x-register
60020 lda xk;x-position in akku
60030 and #$f8;letzten 3 bits ausmaskier
en,
60040 adc ptabl,x;zum lb der zeile addie
ren
60050 sta ptr1;und als lb abspeichern
60060 tya;es wird nur mit 256 x-position
en
60070 adc ptabh,x;gezeichnet.
60080  sta ptr1+1;errechnetes hb in hb d
es pointers
60090 ldx xk;x-position in x-reg. zum zu
griff
60100 lda msk1,x;auf maskentabelle
60110 sta msk;maske abspeichern
60120 ora (ptr1),y;mit grafikbyte 'odern
'
60130 sta (ptr1),y;und abspeichern
60140 ;hier ist carry geloescht, da nie
ein uebertrag auftritt
60170 ldx yk1;zeile 2.punkt
60180 lda xk1
60190 and #$f8;ansteuerung des richtigen
 packens
60200 adc ptabl,x;lb der zeile (0-199)
60210 sta ptr2;in lb des 2.pointers
60220 tya;0-255 also hb des x-wertes = 0
60230 adc ptabh,x;hb der zeile
60240 sta ptr2+1;in hb des 2.pointers
60250 ldx xk1;auslesen der entspr. maske
60260 lda msk1,x;aus der tabelle msk1
60270 sta mskk;und abspeichern in mskk
60280 ora (ptr2),y;maske mit grafik verk
nuepfen
60290 sta (ptr2),y;und abspeichern
60300 rts;ruecksprung zu line
60310 ;
60320 ;insgesammt 96 takte (2 punkte)
60330 ;
```

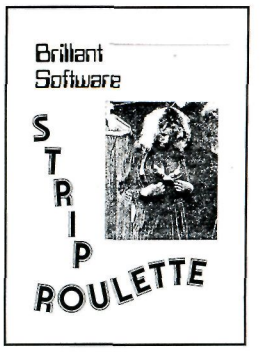
```
100 rem linedemo
110 rem --------
120 rem
130 rem (c) 1986 by jesko schwarzer
140 rem
150 :   a=7*4096
160 rem
170 rem grafik einschalten,
180 rem loeschen
190 rem farben setzen und
200 rem tabellen initialisieren :
210 rem
220 sys a+3
230 rem
240 s=30:p= /180:r=90:xx=160:yy=100
250 fori=0to359steps
260 forj=ito359steps
270 x1=cos(i*p)*r+xx
280 y1=sin(i*p)*r+yy
290 x2=cos(j*p)*r+xx
300 y2=sin(j*p)*r+yy
310 sys a,x1,y1,x2,y2:next:next
320 rem grafik aus
330 poke198,0:wait198,1:poke198,0
340 v=53248:pokev+24,21:pokev+17,27
350 poke56576,3
360 :
```

# Heiße Software zu kühlen Preisen

C64 19,90

C64 19,90

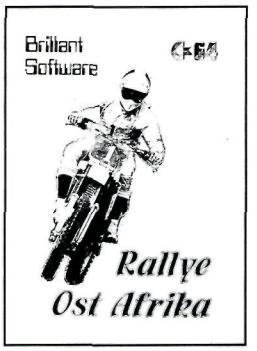
C64 19,90



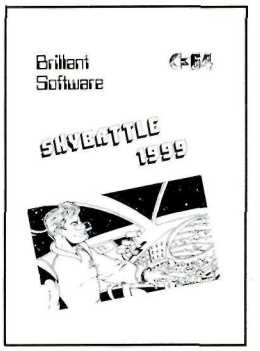

C64 19,90

C64 19,90
Amiga 29,90

C64 19,90

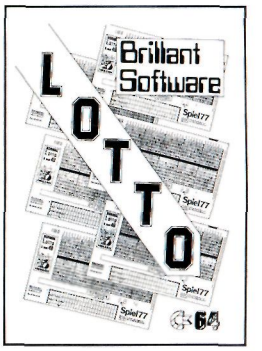



C64 19,90




C64 19,90

C64 24,50
Amiga 29,90




C64 19,90
Amiga 29,90

## Weitere C64 Software:

| | |
|---|---|
| Das RP-System | 49,50 |
| Rollerball | 19,90 |
| Mah Tsu | 19,90 |
| Skybattle 1990 | 19,90 |
| Mission Jonathan | 19,90 |
| War of Wizards | 19,90 |
| Grand National | 19,90 |

**ALLE PROGRAMME NUR AUF DISK**

**C64-Programm des Monats:**
**WAR OF THE WIZARDS**
Die Olympiade der Magier. Mit Zauberkräften und Geschick müssen Sie die voroympischen Disziplinen meistern, um der größte Magier des Jahrtausends, ja aller Zeiten zu werden.
C64 19,90

Das SuperProgramm für die Fans des deutschen Nationalsports Fußball. Alle jemals in der ersten Liga gespielten Ergebnisse seit 1963! Heim- Auswärts- Gesamttabellen, Ewige Liste, Grafische Auswertung, und vieles mehr.

Sämtliche, bisher gezogenen Lottozahlen auf einer Diskette. Alle Möglichkeiten zur statistischen und zufälligen Zahlengenerierung und Analyse.
Dazu ein Lottospiel für die ganze Familie.

## AMIGA Publik-Domain-Service

Alle Fred Fish bis 102 Bestellwert + 3,-DM Porto
1-10 Disketten 6,00 DM pro Stück
ab 11 Disketten 5,50 DM pro Stück

**High Speed Software W. Blanke Postfach 3362 Bad Grund/Harz**

Ich bestelle:

Name: Straße: Wohnort:

☐ Ich bezahle per Nachnahme zzgl. 5,-DM Versandkosten ☐ Verrechnungsscheck liegt bei Unterschrift:

# Roeske's Computer Mag.

## Sofort-Bestellschein für ein persönliches Abonnement

Ich beziehe Roeske's Computer Mag. noch **nicht** per Post und möchte das Magazin für Aufsteiger jetzt abonnieren. Liefern Sie mir deshalb Roeske's Computer Mag. ab der nächsten erreichbaren Ausgabe für die Dauer eines Jahres und weiter bis zur Abbestellung regelmäßig einmal im Monat zum Preis von 68,- DM für 12 Hefte.

Es entstehen keine weiteren Kosten. Lieferung erfolgt frei Haus; Porto und Zustellgebühren trägt der Verlag.

Name _____ Vorname _____

Straße _____

PLZ/Ort _____ Datum/Unterschrift _____

Das Abonnement verlängert sich nur dann zu den dann jeweils gültigen Bestimmungen, wenn es nicht 2 Monate vor Ablauf schriftlich gekündigt wird. Ich bezahle mein Abonnement jährlich im voraus.

Mir ist bekannt, daß ich diese Bestellung innerhalb von 8 Tagen bei der Bestelladresse widerrufen kann. Zur Wahrung der Frist genügt die rechtzeitige Absendung des Widerrufs. Ich bestätige dies durch meine zweite Unterschrift.

Datum/Unterschrift _____

Sie sparen 13,60 DM wenn Sie Roeske's Computer Mag. im Abo beziehen

# Roeske's Computer Mag.

## Bestellschein für ein Geschenk-Abonnement

Ich möchte Abonnement von Roeske's Computer Mag. verschenken. Für dieses Geschenkabonnement zahle ich einschließlich Frei-Haus-Lieferung und MwSt. jährlich 68,- DM (12 Hefte) im voraus. Es entstehen mir keine weiteren Kosten.

**Adresse des Bestellers:**

Name _____ Vorname _____

Straße _____

PLZ/Ort _____ Datum/Unterschrift _____

☐ Das Abonnement verlängert sich nur dann zu den dann jeweils gültigen Bestimmungen, wenn es nicht 2 Monate vor Ablauf schriftlich gekündigt wird.

☐ Das Abonnement ist limitiert auf 12 Hefte.

**Adresse des Abonnement-Empfängers:**

Name _____ Vorname _____

Straße _____

PLZ/Ort _____

Mir ist bekannt, daß ich diese Bestellung innerhalb von 8 Tagen bei der Bestelladresse widerrufen kann. Zur Wahrung der Frist genügt die rechtzeitige Absendung des Widerrufs. Ich bestätige dies durch meine zweite Unterschrift.

Datum/Unterschrift _____

Sie erhalten es monatlich druckfrisch durch Ihren Postboten

Auch als Geschenk ist es ideal!

Bitte ausschneiden oder Fotokopieren und einsenden an:

**Roeske's Computer Mag. ★ Abo-Service ★ Westring 59c ★ D-3440 Eschwege**

# Gewinnen Sie 7500,- DM

**Achtung, Programmierer! Nach wie vor aktuell ist für Sie unser Programmierwettbewerb**
**Das Superprogramm des Monats !**
**Und zum Jahresende können Sie, liebe Leser, die keine Superprogrammierer sind, gewinnen, wenn es gilt, Das Superprogramm des Jahres zu wählen.**

Jeden Monat sehen wir nach, ob unter den Programmeinsendungen ein Super-Programm des Monats dabei ist.
Es kann sich hierbei um ein Spiel, eine Anwendung oder um eine Demo von Grafik, Sound oder sonstigen Fähigkeiten des Computers handeln.
Dieses Programm prämieren wir mit **2500,- DM**.
**Den Sieger stellen wir im Heft und auch auf Video vor.**
Als Prämie für das von den Lesern gewählte Programm des Jahres erhält der Programmierer noch einmal das Doppelte, also zusätzlich **5000,- DM**. Insgesamt sind demnach für das Programm des Jahres 1988 **7500,- DM** zu gewinnen.
Es lohnt sich, wie Sie sehen, uns Ihr bestes Programm zuzusenden.
Aber auch Sie als Leser gehen bei der Wahl nicht leer aus. Interessante und wertvolle Preise werden unter den Einsendern verlost.
**Machen Sie mit!** Es lohnt sich. Selbst wenn Ihr Programm nicht das Programm des Monats wird, es sich aber unter den besten des Jahres befindet, so haben Sie immer noch die Chance, in einem der nächsten Monate dabei zu sein.
Und sollte Ihr Programm nicht zu den besten des Jahres gehören, so winkt doch eine Veröffentlichung, für die wir ein angemessenes, gutes Honorar zahlen!
Der Programmierwettbewerb ist offen für **Amiga** und **C64**-Programme. Der Rechtsweg ist ausgeschlossen.
Senden Sie Ihr Programm so schell wie möglich an die Redaktion. Natürlich sollte es gut ausgetestet, ausreichend dokumentiert und mit Lade- Abtipp- und Bedienungsanleitung versehen sein.

Senden Sie Ihre Programme an:

**Roeske's Computer Mag.**
**Postfach 620**
**Westring 59c**
**3440 Eschwege**

---

Postscheckkonto Nr. des Absenders

**Empfängerabschnitt**

DM | Pf

für Postscheckkonto Nr.
5275 16 - 603
**Lieferanschrift** und **Absender** der Zahlkarte

PLZ Ort

Verwendungszweck

**Programm-Service**

Meine Kunden-Nr.

DM | Pf | für Postscheckkonto Nr. 5275 16 - 603

Absender der Zahlkarte

PSchA | Postscheckkonto Nr. des Absenders | Postscheckteilnehmer

**Zahlkarte/Postüberweisung**

Die **stark umrandeten Felder** sind nur auszufüllen, wenn ein **Postscheckkontoinhaber** das Formblatt als **Postüberweisung** verwendet (Erläuterung s. Rücks.)

DM | Pf | (DM-Betrag in Buchstaben wiederholen)

für Monika Roeske

in 3440 Eschwege

Postscheckkonto Nr.
5275 16 - 603
Postscheckamt
Frankfurt am Main

Ausstellungsdatum Unterschrift

Für Vermerke des Absenders

Postscheckkonto Nr. des Absenders

**Einlieferungsschein/Lastschriftzettel**

DM | Pf

für Postscheckkonto Nr. 5275 16 - 603

Postscheckamt Ffm

für Monika Roeske
Westring 59c
in 3440 Eschwege

Postvermerk

# Inserenten dieser Ausgabe


| | |
|---|---|
| **Atlantis** | **80** |
| **CAS Computer** | **52** |
| **CSJ Computersoft** | **2** |
| **High Speed Software** | **75** |
| **Rainbow Arts** | **25** |
| **Video Loft** | **35** |


# Impressum


**Herausgeber:**
Monika Roeske

**Chefredakteur:**
Ralph Roeske

**Redaktion:**
Carsten Borgmeier
Horst Franke
Garry Glendown
Michael Kaiser
Sascha Schädel

**Manuskripte und Programme**
Manuskripte und Programme werden vom Verlag gerne angenommen.
Sie müssen frei von Rechten Dritter sein. Geben Sie bitte an, wenn Sie Ihre Einsendung auch an anderer Stelle zur Veröffentlichung oder gewerblichen Nutzung angeboten haben.
Mit der Einsendung an den Verlag gibt der Verfasser die Zustimmung zum Abdruck in den Zeitschriften des Roeske Verlags, sowie zur Vervielfältigung der Programme auf Datenträgern.
Mit der Einsendung von Bauanleitungen gibt der Einsender die Zustimmung zum Abdruck in den Roeske Verlag Publikationen und dazu, daß der Verlag Geräte und Bauteile nach der Bauanleitung herstellen läßt und vertreibt oder durch Dritte vertreiben läßt.
Honorare nach Vereinbarung.
Für unverlangt eingesandte Manuskripte und Listings wird keine Haftung übernommen.
Für gewünschte Rücksendung frankierten beschrifteten Rückumschlag beilegen.

**Anzeigenleitung:**
Monika Roeske

**Vertrieb:**
Districenter, Bad Schwalbach

**Verantwortlich:**
Redaktionell: Ralph Roeske
Anzeigen: Monika Roeske

**Anschrift:**
**Roeske's Computer Mag.**
**c/o Roeske Verlag**
**Westring 59c**
**D-3440 Eschwege**
Telefon: 05651-8777
Telefax: 05602-7711



**Urheberrecht:**
Alle in Roeske's Computer Mag erschienen Beiträge sind urheberrechtlich geschützt. Reproduktionen und Erfassung auf Datenträgern aller Art bedürfen der schriftlichen Zustimmung des Verlags.

**Haftung:**
Für Fehler in Listings und Bauanleitungen übernimmt der Verlag keine Haftung. Verwendung der angebotenen Soft- und Hardware erfolgt auf eigenes Risiko des Nutzers.



Einlieferungsschein/Lastschriftzettel
(nicht zu Mitteilungen an den Empfänger benutzen)

Gebühr für die Zahlkarte
(wird bei der Einlieferung bar erhoben)

bis 10 DM — — — 90 Pf
über 10 DM (unbeschränkt) 1,50 DM

Bei Verwendung als Postüberweisung gebührenfrei

Bedienen Sie sich der Vorteile eines eigenen Postgirokontos

Auskunft hierüber erteilt jedes Postamt

Mitmeis für Postgirokontoinhaber:
Dieses Formblatt können Sie auch als Postüberweisung benutzen, wenn Sie die stark umrandeten Felder zusätzlich ausfüllen. Die Wiederholung des Betrages in Buchstaben ist dann nicht erforderlich. Ihren Absender (mit Postleitzahl) brauchen Sie nur auf dem linken Abschnitt anzugeben.
1. Abkürzung für den Namen Ihres Postgiroamts (PGiroA) siehe unten
2. Im Feld »Postgirotelnehmer« genügt Ihre Namensangabe
3. Die Unterschrift muß mit der beim Postgiroamt hinterlegten Unterschriftsprobe übereinstimmen
4. Bei Einsendung an das Postgiroamt bitte den Lastschriftzettel nach hinten umschlagen

Für Mitteilungen an den Empfänger

Bestellung Programm-/Hardware -Service | Wichtig: Lieferanschrift (Rückseite) nicht vergessen!

| Bestell-Nr | Anzahl | x Einzelpreis | = Gesamtpreis |
|---|---|---|---|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| Summe bitte auf Vorderseite übertragen | | | Gesamtsumme |

Abkürzungen für die Ortsnamen der PGiroÄ:

| | | | |
|---|---|---|---|
| Bln W = | Berlin West | Kln = | Köln |
| Dtmd = | Dortmund | Lhfn = | Ludwigshafen am Rhein |
| Esn = | Essen | Mch = | München |
| Ffm = | Frankfurt am Main | Nbg = | Nürnberg |
| Hmb = | Hamburg | Sbr = | Saarbrücken |
| Han = | Hannover | Stgt = | Stuttgart |
| Krh = | Karlsruhe | | |

Feld für postdienstliche Zwecke

# Disketten- & Video-Service

## So tippen Sie die C64-Listings richtig ein.

Wir haben als Reaktion auf die Art, wie wir im letzten Heft die C-64 Listings abdruckten, einige gute Tips und konstruktive Kritik erhalten.
Es sieht so aus, als wenn sich der Leser von Computerzeitschriften an die seltsamen Listings mit "Space" und "blk" so sehr gewöhnt hat, daß er sich ungern umstellt.
Dem tragen wir Rechnung, denn Roeske's Computer Mag. ist die Zeitung für Sie und Sie entscheiden letztendlich, wie Ihnen das Abtippen am leichtesten fällt.
Ab sofort sind die C-64-Listings also in leicht verständlicher Form abgedruckt.
Dazu einige Hinweise: Alles, was beim C-64 unter der Kategorie Steuerzeichen läuft, befindet sich innerhalb von zwei invers gedruckten eckigen Klammern.
Zum Beispiel SPACE für Leertaste, HOME für Cursor home oder shCLR für Clear Screen.
Das klein gedruckte sh bedeutet Shift-Taste gedrückt halten und in obigem Beispiel CLR drücken.
Die Control-Taste wird mit ct bezeichnet und die Commodore-Taste mit c=. Auch finden Sie bei uns nicht grey 2 oder irgend eine Farbangabe und müssen dann die entsprechende Taste suchen. Wenn schon, dann richtig, so sagten wir uns und drucken nun aus, welche Taste mit welcher Kombination, ob c= oder ct!
Wenn ein Zeichen mehr als einmal hintereinander auftaucht, so wird weiß auf schwarzem Kästchen angezeigt, wie häufig die entsprechende Taste zu drücken ist, um die Zeichen darzustellen. GOTO 2000 wird somit zu goto 2«3x»0
Damit 1 und l nicht verwechselt werden können, drucken wir das große L anstelle des kleinen l; aber nicht im Basic-Befehl.

## Disketten-Service

Wie jeden Monat, so sind auch diesmal alle im Heft enthaltenen Programme auf den Service-Disketten erhältlich. Auf diesen Disketten befinden sich nicht nur die Source-Programme, sondern, wenn es sich um Assembler oder C-Programme handelt auch die **assemblerte**, bzw. **compilierte** Version!
Sofern Basic-Programme durch compilieren mit einem Basic-Compiler entscheidend verbessert werden, befinden sich auch compilierte Versionen von Basic-Programmen auf der Service-Disk.
**Bestellen Sie bitte nur mit nebenstehender Zahlkarte oder per Überweisung. Aus Zeitgründen können wir keine Nachnahmen mehr ausführen!**

**Vergessen Sie bitte Ihren Absender nicht.**

Die Bestellnummer ist jeweils die auf dem Titel des Heftes stehende Ausgabenummer, z.B. diesen Monat 1/88 - und fügen Sie bitte ein **A für Amiga** oder **C für C64** an.

| | |
|---|---|
| **Amiga 1/88** | **20,00 DM** |
| **C/64 1/88** | **17,50 DM** |
| **Amiga & C64 1/88** | **32,50 DM** |

## Nachliefer-Service von Zeitschriften und Disketten.

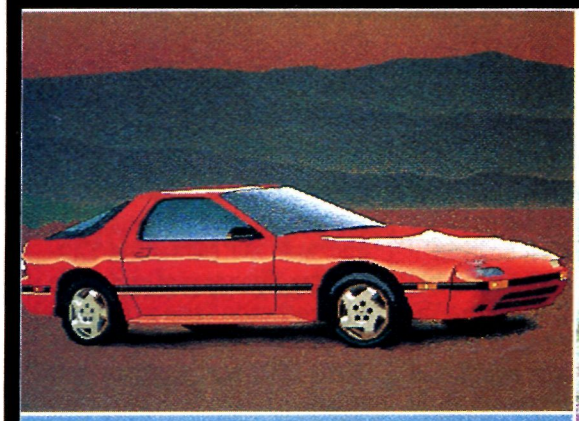
Disketten und Zeitschriften sind ab Heftfolge 11/87 erhältlich solange der Vorrat reicht.
Die Preise lauten wie folgt:

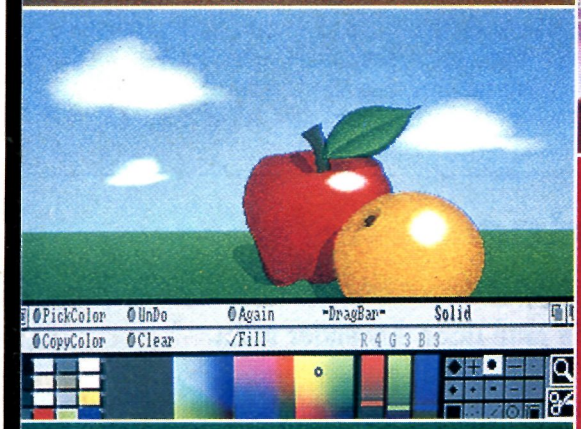
| | | | |
|---|---|---|---|
| **Amiga 11/87** | **15,00 DM** | **Amiga 12/87** | **20,00 DM** |
| **C64 11/87** | **15,00 DM** | **C64 12/87** | **20,00 DM** |
| **Amiga & C64 11/87** | **27,50 DM** | **Amiga & C64 12/87** | **35,00 DM** |

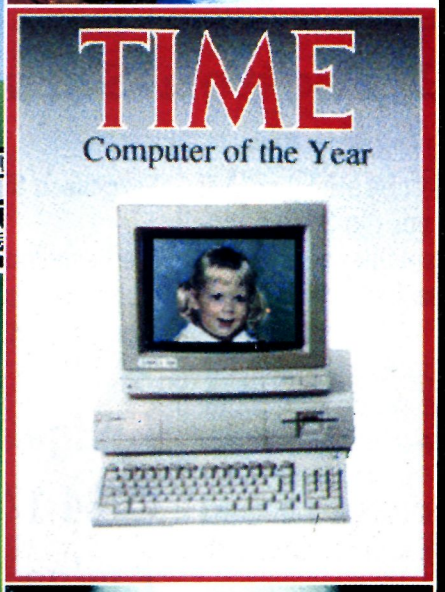
Zeitschriften kosten 6,80 DM incl. Porto pro Exemplar.
Bestellungen aus dem Ausland nur gegen bar oder Euroscheck.

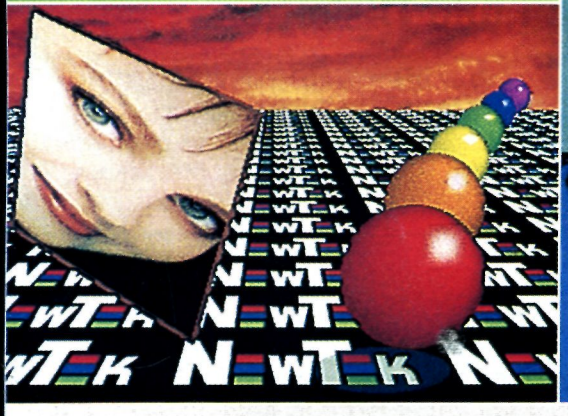
## Video-Service

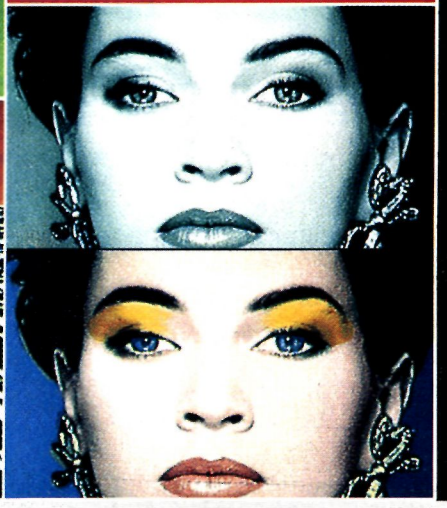
Beginnend mit dieser Ausgabe starten wir einen Video-Service, um Ihnen die Möglichkeit zu bieten, sich noch besser zu informieren.
Es gibt zwei Möglichkeiten, diesen Service zu nutzen:
**1. Mieten**: Sie zahlen 15,00 DM mit nebenstehender Zahlkarte ein. Innerhalb der nächsten 21 Tage nach Zahlungseingang senden wir Ihnen eine Video-Kassette mit dem bestellten Programm zu. Diese senden Sie bitte innerhalb der nächsten 48 Stunden nach Eingang bei Ihnen wieder frankiert und in Original-Verpackung zurück.
**2. Kaufen**: Sie zahlen 39,00 DM auf unser, auf der nebenstehenden Zahlkarte bezeichnetes Konto und Sie erhalten Ihr Video auf fabrikneuer Kassette.
Bedenken Sie bitte, daß in diesem Preis kein Recht auf Weitervermietung enthalten ist.
Videotheken forden bitte unsere aktuellen Preise und Lieferbedingungen an.









# DIGI PAINT

**Das erste Malprogramm, das die 4.096 Farben des Amigas voll ausnutzt.**
Jetzt gibt es Digi-Paint in einer deutschen Version.

Die Vorteile von Digi-Paint sind:

- voll IFF und Digi-View kompatibel
- 4.096 Farben zur gleichen Zeit
- exaktes Ausschneiden von Bildteilen und Wiedereinsetzen an anderer Stelle
- komplett in Assembler programmiert
- einfache Bedienung von Farben und Menüleiste
- volles Abspeichern und Laden von IFF-Bildern
- zwölf verschiedene Malmoden einschließlich Vermischen, Färben und Schattieren
- Kreise, Elipsen, Rechtecke und andere Zeichenhilfen
- es arbeitet im 320x200 und im hi-res 320x400 HAM-Modus
- zum detaillierten Arbeiten kann man Ausschnitte vergrößern
- Effekte wie Verdoppeln, Halbieren, Spiegeln und mehr

Preis DM **138,–**

Alle hier gezeigten Bilder wurden mit Digi-Paint erstellt. Digi-Paint ist natürlich Digi-View-kompatibel.

Die deutschen Digi-Paint- und Digi-View-Pal-Versionen erhalten Sie von Ihrem Commodore-Händler oder direkt von ATLANTIS bzw. CompuStore.

NewTek INCORPORATED

ATLANTIS
Soft- und Hardware GmbH
Dunantstraße 53 · 5030 Hürth
Telefon 0 22 33 / 4 10 81

&

CompuStore
Handelsgesellschaft mbH für Hard- und Software
Fritz-Reuter-Straße 6 · 6000 Frankfurt/Main 1
Telefon 069/567399